

PROBLEME

1. Verifică dacă un număr natural n este prim.
2. Se consideră o aplicație care realizează managementul quiz-urilor efectuate de un student. Punctajul unui quiz este o valoare de la 0 la 30. Lista de quiz-uri poate să conțină până la 100 quiz-uri. Lista de funcționalități conține:

F12. Să se determine numărul de quiz-uri (**maxCounter**) pentru care studentul a obținut cel mai mare punctaj al său. Dacă nu s-au efectuat quiz-uri atunci **maxCounter** este 0. Dacă punctajul obținut pentru fiecare quiz este 0 atunci **maxCounter** este 0.

Clasa Service conține o metodă pentru această funcționalitate, care primește la intrare o listă cu valori întregi ce reprezintă punctajele testelor efectuate și returnează o valoare întreagă ce reprezintă **maxCounter**. Pentru situațiile excepționale se returnează un mesaj de eroare.

CERINȚE

- I. Să se specifice metodele descrise de funcționalitățile de mai sus ($X, Z, \varphi(X), \Psi(X, Z)$).
- II. Să se proiecteze cazuri de testare folosind tehnicile de testare black-box ECP și BVA.
- III. Să se determine acoperirea pentru tehnicile de testare folosite.

Etape de aplicare a ECP:

1. identificarea ECs valide/non-valide pentru datele de intrare/ieșire;
2. proiectarea TCs pentru ECs identificate;

Etape de aplicare a BVA:

1. identificarea condițiilor BVA pentru ECs valide existente;
2. proiectarea TCs pentru condițiile BVA identificate.

ECP. Proiectarea cazurilor de testare. Reguli 1

1. dacă o condiție de intrare precizează apartenența la un interval de valori $[a,b]$:
 - ==> 1 EC validă, 2 EC non-valide;
 - E.g.: luna, o valoare intervalul $[1, 12]$;
2. dacă o condiție de intrare precizează o mulțime finită de valori de intrare:
 - ==> 1 EC validă pentru fiecare valoare, 1 EC non-validă;
 - E.g.: $\text{tip curs} \in \text{CourseType} = \{\text{optional}, \text{obligatoriu}, \text{facultativ}\}$;
 - 1 EC validă pentru fiecare element din CourseType:
 - $\text{EC}_1: \{\text{optional}\}, \text{EC}_2: \{\text{obligatoriu}\}, \text{EC}_3: \{\text{facultativ}\} \implies 3 \text{ ECs valide}$;
 - 1 EC non-validă: $\text{EC}_4: M = \{e \mid e \notin \text{CourseType}\}$;

ECP. Proiectarea cazurilor de testare. Reguli 2

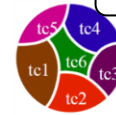
3. dacă o condiție de intrare precizează numărul de valori:
 - ==> 1 EC validă, 2 EC non-valide;
 - E.g.: "de la 1 până la 5 studenți";
 - 1 EC validă: $\text{EC}_1: D=[1,5]$; 2 EC non-valide: $\text{EC}_2: \text{nici un student}$; $\text{EC}_3: \text{mai mult de 5 studenți}$;
4. dacă o condiție de intrare precizează o situație de tipul "must be":
 - ==> 1 EC validă, 1 EC non-validă.
 - E.g.: "primul caracter din parolă trebuie să fie un simbol numeric";
 - 1 EC validă:
 - $\text{EC}_1: \text{primul caracter este un simbol numeric}$;
 - 1 EC non-validă:
 - $\text{EC}_2: \text{primul caracter nu este un simbol numeric}$.

ECP. Algoritm

3

- Algoritm de aplicare a ECP (*identificarea ECs și proiectarea TCs*):

1. se identifică clasele de echivalență pe baza condițiilor de intrare/ieșire;
2. se clasifică clasele de echivalență în:
 - **valide** – formate din datele de intrare/ieșire valide pentru program;
 - **non-valide** – formate din datele de intrare/ieșire eronate, corespunzătoare tuturor celorlalte stări ale condiției de intrare/ieșire.
3. se asociază un identificator unic fiecărei clase de echivalență (e.g., EC₁, EC₂, etc.);
4. **cât timp** (*nu au fost descrise cazuri de testare pentru toate clasele de echivalență valide/non-valide*):
 - **scrie** (*un nou caz de testare care corespunde la cât mai multe clase de echivalență valide încă neacoperite*);
 - **scrie** (*un nou caz de testare care corespunde doar uneia dintre clasele de echivalență de non-valide încă neacoperite*).



ECP. Acoperirea testării ECs

4

- calculul acoperirii (*engl. coverage*) testării ECs pentru tehnica de testare ECP:

$$\text{Acoperirea ECs} = \frac{\text{numărul de ECs testate}}{\text{numărul de ECs identificate}} \times 100$$

BVA. Proiectarea cazurilor de testare. Reguli

5

1. **dacă o condiție de intrare/ieșire precizează apartenența la un interval de valori [a,b]:**
 - ==> **cazuri de testare pentru:**
 - (1) **condiții BVA valide** - **limitele intervalului** (e.g., a, a+1; b-1, b);
 - (2) **condiții BVA non-valide** - valori aflate în afara intervalului (e.g., a-1, b+1);
2. **dacă o condiție de intrare/ieșire precizează o mulțime de valori ordonată:**
 - ==> **cazuri de testare pentru:**
 - (1) **condiții BVA valide** - **primul și ultimul** element din **mulțime**;
 - (2) **condiții BVA non-valide** - **valoarea** imediat mai mică decât cea mai mică valoare din **mulțime** și valoarea imediat mai mare decât cea mai mare valoare în mulțime;
3. **dacă o condiție de intrare/ieșire precizează numărul de valori (e.g., "de la 1 până la 5 studenți"):**
 - ==> **cazuri de testare pentru:**
 - (1) **condiții BVA valide** - **numărul** minim și maxim de valori, i.e., 1 și 5;
 - (2) **condiții BVA non-valide** - valoarea imediat mai mică și imediat mai mare, i.e. 0 și 6;



Condiții BVA. Sumar

6

Tip ECs	Există limite
interval de valori	da
număr de valori	da
mulțime valori neordonate	nu
mulțime valori ordonate	da
valoare "must be"	nu
secvență	da
ECs dependente	da
variabile multiple dependente	nu

8

BVA. Acoperirea testării condițiilor BVA

- calculul acoperirii (*engl. coverage*) testării condițiilor BVA:

$$\text{Acoperirea BVAs} = \frac{\text{numărul de condiții BVA testate}}{\text{numărul de condiții BVA identificate}} \times 100$$

BVA. Algoritm

7

- Algoritm de aplicare a BVA (*identificarea condițiilor BVA și proiectarea TCs*):

1. se identifică limitele tuturor ECs valide de intrare/ieșire;
2. se scriu condiții BVA pentru fiecare limită a fiecărei EC identificate, astfel încât:
 - valoarea să fie sub limită (mai mică decât limita), e.g., x < 2;
 - valoarea să fie pe limită (egală cu limita), e.g., x = 2;
 - valoarea să fie deasupra limitei (mai mare decât limita), e.g., x > 2;
3. se clasifică condițiile BVA în
 - **valide** – corespund unor date de intrare/ieșire valide pentru program;
 - **non-valide** – corespund unor date de intrare/ieșire non-valide pentru program.
4. se asociază un identificator unic fiecărei condiții BVA (e.g., c1, c2, etc.);
5. **cât timp** (*nu au fost descrise cazuri de testare pentru toate condițiile BVA valide/non-valide*):
 - **scrie** (*un caz de testare nou, care corespunde la cât mai multe condiții BVA valide încă neacoperite*);
 - **scrie** (*un caz de testare nou, care corespunde doar uneia dintre condițiile BVA non-valide încă neacoperite*).

