**PROBLEME**

I. Să se specifice metodele asociate functionalităţilor descrise mai jos (X, Z, φ(X), Ψ(X, Z)).
II. Să se proiecteze cazuri de testare folosind tehnicile de testare black-box ECP şi BVA.
III. Să se determine acoperirea pentru tehnicile de testare folosite.

1. Verifică dacă un număr natural n este prim.
2. O aplicaţie 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 conţină până la 100 quiz-uri. În lista de funcţionalităţi este precizat:
   **F12.** Să se determine numărul de quiz-uri (**maxCounter**) pentru care studentul a obţinut cel mai mare punctaj al său. Dacă nu s-au efectuat quiz-uri atunci **maxCounter** este 0. Dacă punctajul obţinut pentru fiecare quiz este 0 atunci **maxCounter** este 0. Pentru pentru alte situaţii se returnează un mesaj de eroare.
   Clasa Service conţine o metodă pentru această funcţionalitate, care primeşte la intrare o listă cu valori întregi ce reprezintă punctajele testelor efectuate şi returnează o valoare întreagă ce reprezintă **maxCounter**.

**Etape de realizare ECP şi BVA:**

1. **identificarea ECs valide/non-valide, pentru datele de intrare/ieşire (slide 1, slide 2);**
2. **proiectarea TCs pentru ECs identificate (slide 3, punctul 3 şi 4);**
3. **calculul acoperirii pentru ECP (slide 4);**
4. **identificarea condiţiilor BVA pentru ECs valide identificate (slide 5, slide 6);**
5. **proiectarea TCs pentru condiţiile BVA identificate (slide 7, punctul 3 şi 4);**
6. **calculul acoperirii pentru BVA (slide 8).**

## 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.: `tip curs ∈ CourseType = {opțional, obligatoriu, facultativ};`
  - 1 EC validă pentru fiecare element din `CourseType`:
    - **$EC_1$:** {opțional}, **$EC_2$:** {obligatoriu}, **$EC_3$:** {facultativ} ===> 3 ECs valide;
  - 1 EC non-validă: **$EC_4$:** M= {*e* | *e* ∉ `CourseType`};

## ECP. Proiectarea cazurilor de testare. Regu 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: **$EC_1$:** D=[1,5]; 2 EC non-valide: **$EC_2$:** nici un student; **$EC_3$:** 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:
    - **$EC_1$:** primul caracter este un simbol numeric;
  - 1 EC non-validă:
    - **$EC_2$:** primul caracter nu este un simbol numeric.

## ECP. Algoritm

3


- Algoritm de aplicare a ECP (*identificarea ECs şi proiectarea TCs*):
  1. se identifică clasele de echivalenţă pe baza condiţiilor de intrare/ieşire;
  2. se clasifică clasele de echivalenţă în:
     - valide – formate din datele de intrare/ieşire valide pentru program;
     - non-valide – formate din datele de intrare/ieşire eronate, corespunzătoare tuturor celorlalte stări ale condiţiei de intrare/ieşire.
  3. se asociază un identificator unic fiecărei clase de echivalenţă (e.g., $EC_1$, $EC_2$, etc.);
  4. `câttimp` *(nu au fost descrise cazuri de testare pentru toate clasele de echivalenţă valide/non-valide):*
     - `scrie` *(un nou caz de testare care corespunde la cât **mai multe clase de echivalenţă valide** încă neacoperite);*
     - `scrie` *(un nou caz de testare care corespunde **doar uneia dintre clasele de echivalenţă 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 condiţie de intrare/ieşire precizează apartenenţa la un interval de valori [a,b]:**
- ==> cazuri de testare pentru:
  - (1) condiţii BVA valide - limitele intervalului (e.g., a, a+1; b-1, b);
  - (2) condiţii BVA non-valide - valori aflate în afara intervalului (e.g., a-1, b+1);

2. **dacă o condiţie de intrare/ieşire precizează o mulţime de valori ordonată:**
- ==> cazuri de testare pentru:
  - (1) condiţii BVA valide - primul şi ultimul element din mulţime;
  - (2) condiţii BVA non-valide – valoarea imediat mai mică decât cea mai mică valoare din mulţime şi valoarea imediat mai mare decât cea mai mare valoare în mulţime;

3. **dacă o condiţie de intrare/ieşire precizează numărul de valori (e.g., "de la 1 până la 5 studenţi"):**
- ==> cazuri de testare pentru:
  - (1) condiţii BVA valide – numărul minim şi maxim de valori, i.e., 1 şi 5;
  - (2) condiţii BVA non-valide – valoarea imediat mai mică şi imediat mai mare, i.e. 0 şi 6;

## Condiţii BVA. Sumar

6

| Tip ECs | Există limite |
|---|---|
| interval de valori | da |
| număr de valori | da |
| mulţime valori neordonate | nu |
| mulţime valori ordonate | da |
| valoare "must be" | nu |
| secvenţă | da |
| ECs dependente | da |
| variabile multiple dependente | nu |

## BVA. Acoperirea testării condiţiilor BVA

8

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

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

## BVA. Algoritm

7

- Algoritm de aplicare a BVA (*identificarea condiţiilor BVA şi proiectarea TCs*):
  1. se identifică limitele tuturor ECs valide de intrare/ieşire;
  2. se scriu condiţii 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 condiţiile BVA în
     - valide – corespund unor date de intrare/ieşire valide pentru program;
     - non-valide – corespund unor date de intrare/ieşire non-valide pentru program.
  4. se asociază un identificator unic fiecărei condiţii BVA (e.g., c1, c2, etc.);
  5. `câttimp` *(nu au fost descrise cazuri de testare pentru toate condiţiile BVA valide/non-valide):*
     - `scrie` *(un caz de testare nou, care corespunde la **cât mai multe condiţii BVA valide** încă neacoperite);*
     - `scrie` *(un caz de testare nou, care corespunde **doar uneia dintre condiţiile BVA non-valide** încă neacoperite).*