

ANEXA

Modalitatea de realizare a selecției aleatorii a candidaților aflați în situație de egalitate

În vederea asigurării transparenței, imparțialității și verificabilității procesului de selecție, ordinea finală a candidaților se stabilește printr-un algoritm public, utilizând ca sursă de aleatorietate un hash extras din blockchain-ul Ethereum.

1. Lista inițială

Selecția se realizează asupra listei de candidați, în ordinea inițială, publicată înainte de efectuarea extragerii.

Lista inițială utilizată la selecție este reprezentată de tabelul de clasificare a candidaților, publicat la adresa <https://www.csm1909.ro/>, la secțiunea aferentă examenului de capacitate.

2. Sursa de aleatorietate

Pentru generarea rezultatului se utilizează **hash-ul primului bloc Ethereum generat după data și ora:**

Data:19.06.2026 **Ora:** 14.00.00-ora locală a României (București, UTC+3)

Hash-ul va fi preluat din exploratorul public Ethereum: <https://etherscan.io/>

Orice persoană interesată poate verifica ulterior blocul utilizat și valoarea hash-ului asociată.

3. Algoritmul de selecție

Algoritmul utilizat este public și disponibil la: <https://github.com/ifep-ro/select>

Procesul se desfășoară astfel:

1. Se preia hash-ul primului bloc Ethereum generat după data și ora menționate.
2. Din hash se extrag primii 16 octeți (128 biți).
3. Aceștia sunt transformați într-o valoare numerică utilizată drept sămânță (*seed*) pentru generatorul de numere pseudo-aleatoare.
4. Lista inițială a candidaților este amestecată (*shuffle*) utilizând *seed* de la pct. 3.
5. Ordinea rezultată constituie **ordinea de prioritate aleatorie** a candidaților și ***este utilizată exclusiv pentru departajare în situațiile în care criteriile prevăzute de regulamentul de concurs nu permit stabilirea unei ierarhii distincte între candidații aflați în situație de egalitate.***

4. Codul sursă utilizat

```
var hashBytes = Convert.FromHexString(hash);
var seedBytes = hashBytes.Take(16).ToArray();
var bigSeed = new BigInteger(seedBytes, true, true);
var seed = (int)(bigSeed % int.MaxValue);
var rnd = new Random(seed);
var shuffled = identifiers
    .OrderBy(_ => rnd.Next())
    .ToList();
return shuffled.Take(request.Count).ToList();
```

5. Utilizarea rezultatului

Rezultatul generat de algoritm **nu reprezintă clasamentul final al candidaților și nu înlocuiește criteriile de evaluare stabilite prin procedura de selecție.**

Ordinea de prioritate generată aleator este *utilizată doar în situațiile în care doi sau mai mulți candidați nu pot fi departajați pe baza criteriilor prevăzute de art. 39 din Regulamentul de concurs*

În aceste situații, ordinea rezultată din mecanismul descris în prezenta anexă reprezintă criteriul final de departajare, utilizat pentru stabilirea priorității între candidații aflați la egalitate.

6. Verificabilitate

Mecanismul este complet verificabil și reproductibil. Orice persoană poate reproduce rezultatul utilizând:

- lista inițială a candidaților publicată anterior extragerii;
- hash-ul blocului Ethereum utilizat;
- algoritmul public disponibil la adresa <https://github.com/ifep-ro/select>.

În aceste condiții, aceleași date de intrare vor genera întotdeauna aceeași ordine de prioritate, fără posibilitatea modificării ulterioare a rezultatului.