

Tema 1 POO

VoteSmart – Platformă de Votare Online

Responsabili: Laur Neagu

Introducere

Scopul acestei teme este dezvoltarea unei platforme de vot online simplificată pentru a simula interacțiunile prin intermediul claselor și obiectelor. Această temă vă va ajuta să înțelegeți și să aplicați conceptele deja învățate ale programării orientate pe obiect (OOP).

Scheletul de cod furnizat include și un set de teste publice prin care puteți să vedeți, în timp ce dezvoltați, procentul de acoperire al soluției voastre raportat la cerințele temei. Vă recomandăm să studiați testele publice furnizate pentru a vedea formatul de ieșire așteptat și cazuri limită ale implementărilor.

Această temă abordează o metodologie de tip Test-Driven Development (TDD)¹, prin care aveți la dispoziție “așteptările” din partea software-ului, iar voi va trebui să „aliniați” dezvoltarea acestuia.

Reguli generale

1. Nu este permis să modificați numele fișierului sursă unde este metoda main(), folosiți numele furnizat în schelet, și anume Tema1.java
2. Nu este permis să modificați bucata de cod din metoda main() ce face parte din scheletele aplicației. Aceasta va folosi la testare.
3. Nu este permis să modificați nimic (linii de cod, fișiere, foldere) ce se află în directorul src/test, din scheletul de cod.
4. Nu este permis să implementați tema folosind EXCLUSIV generatoare de cod (Github Copilot / ChatGPT / etc.). La verificarea de plagiat, se va măsura similaritatea codului vostru cu generatoarele de cod.
5. Vă puteți instala și configura Gradle (nu este obligatoriu)², și rula aceste teste pe mediile voastre locale (rulând comanda: gradle test). La încărcarea codului pe GitHub Classroom (rulând comanda: git push), aceste teste vor rula automat și veți vedea punctajul. Important de menționat că aceste teste publice sunt doar un set parțial de teste redactate pentru această temă. Tema include și un set de teste private ce se vor rula ulterior.

Descriere problemă

Tema își propune implementarea unui sistem interactiv de gestionare a procesului electoral. Alegerile se desfășoară în mai multe circumscripții electorale, fiecare circumscripție având proprii candidați, votanți și statistici. Aplicația va permite gestionarea acestora și va monitoriza voturile pentru a preveni acțiunile frauduloase.

O sesiune de alegeri este împărțită în mai multe subdiviziuni teritoriale numite circumscripții. După ce va începe alegerea se vor putea gestiona votanții pentru o circumscripție, ceea ce include adăugare,

¹ <https://testdriven.io/test-driven-development>

² <https://gradle.org/>

ștergere și listare. Se vor putea înregistra voturi, pe baza CNP-ului votantului și al candidatului. Există noțiunea de încercare de fraudă, dacă un votant încearcă să voteze mai mult de o singură dată, sau dacă nu este înregistrat în circumscripția în care se face votul.

După ce se încheie alegerile se vor putea face rapoarte voturi per circumscripție / național și se vor putea lista toate fraudele raportate. De asemenea se va putea realiza o analiză detaliată per circumscripție / întreaga țară.

Implementați clasele Alegeri, Circumscripție, Persoană, Candidat, Votant, Vot, Fraudă, Analiză ș.a. cu atributele și metode adecvate. Asigurați-vă că datele sunt încapsulate corespunzător și că modificatorii de acces sunt utilizați pentru a restricționa accesul la membrii claselor.

Reguli implementare temă

- O sesiune de alegeri poate începe DOAR dacă curent se află în stagiul de NEÎNCEPUT.
- Datele precum CNP și vârstă trebuie validate înainte de a adăuga un votant sau candidat.
- Dacă există deja o circumscripție cu același nume sau o persoană cu același CNP, se va afișa o eroare.
- Dacă se vor realiza operații pe obiecte inexistente, se va afișa o eroare.
- O sesiune de alegeri se poate termina DOAR dacă curent se află în stagiul de ÎN_CURS.

Tema va fi implementată sub forma unei aplicații interactive Java de tip consolă, și va accepta o serie de comenzi de la tastatură. Aceasta va rula până când utilizatorul va alege să o termine.

Următoarele comenzi vor trebui procesate în tema voastră. Documentația completă a acestor comenzi, cu răspunsurile așteptate în funcție de apel, este disponibilă în documentul Tema1-DocumentațieComenzi-2024.pdf.

0. Creare alegeri
1. Pornire alegeri
2. Adăugare circumscripție
3. Eliminare circumscripție
4. Adăugare candidat în alegeri
5. Eliminare candidat din alegeri
6. Adăugare votant în circumscripție
7. Listarea candidaților din alegeri
8. Listarea votanților dintr-o circumscripție
9. Votare
10. Oprire alegeri
11. Creează raport voturi per circumscripție
12. Creează raport voturi pe plan național
13. Analiză detaliată per circumscripție
14. Analiză detaliată pe plan național
15. Raportare fraude
16. Șterge alegeri
17. Listare alegeri
18. Ieșire

Criterii de Evaluare

- Încapsularea corectă a datelor în cadrul claselor.
- Demonstrarea interacțiunii între obiecte cum ar fi procesul de votare și realizarea rapoartelor.
- Utilizarea corectă a abstractizării pentru a ascunde detaliile de implementare.

Punctaj

DOMENIU	TASK	PUNCTAJ	TOTAL
Test	<i>Suită de teste rulată automat</i>	80 p	80 PUNCTE
Cod	<i>Folosirea conceptelor de POO studiate</i> <i>Folosirea structurilor de date studiate</i> <i>Lizibilitate cod, comentarii (acolo unde este cazul)</i>	5 p 5 p 5 p	15 PUNCTE
README	<i>Redactarea unui fișier Readme în care să se detalieze implementarea temei (200-500 cuvinte)</i>	5 p	5 PUNCTE

Bonus

Ce alte cazuri limită ați mai trata în această aplicație (minim 3 cazuri, a se descrie în README)?

– 5 PUNCTE

Cum ați refactoriza comenzile și răspunsurile din aplicație (minim 3 propuneri, a se descrie în README)?

– 5 PUNCTE

Totodată, pentru a vă maximiza punctajul în cadrul acestei teme, încercați să acoperiți cât mai multe dintre cazurile întoarse de fiecare comandă, și din lanțuri de comenzi succesive.