

A. Calitatea codului

1. Conformitatea cu stilul de cod Java. Claritatea codului:

- Verificați dacă sunt respectate convențiile de denumire Java si este pastrata consistenta notatiilor;
- Asigurați-vă că sunt respectate regulile de formatare, cum ar fi indentarea și utilizarea corectă a acoladelor;
- Asigurați-vă că numele variabilelor și metodelor sunt descriptive.
- Asigurați-vă că este un cod clar și ușor de înțeles;
- Verificați dacă sunt comentarii adecvate pentru a explica blocurile de cod complexe sau algoritmi;

2. Gestionarea excepțiilor:

- Asigurați-vă că sunt gestionate excepțiile adecvat în cod;
- Verificați dacă excepțiile sunt înregistrate corespunzător sau transmise mai departe în mod adecvat.

3. Folosirea corecta a conceptelor OOP:

- Relatiile dintre clase sunt modelate corect? (agregare vs mostenire etc)
- Este folosit polimorfismul (static, dinamic, parametric)? in ce fel?
- Identificati design patterns din cod; sunt corect folosite?

4. Verificarea metricilor OOP:

- Verificați respectarea metricilor definite in fisierul "metriciPOO.doc"

5. Verificarea eficientei si performantei

- Codul are o complexitate optima?
- Exista alocari inutile de resurse?

B. Corectitudinea functionala a codului

6. Corectitudine funcțională. Dependențe și biblioteci

- Asigurați-vă că codul funcționează corect și rezolvă problema pentru care a fost scris;
- Verificați dacă sunt folosite corect și corespunzător orice biblioteci sau dependențe externe.

C. Sumarul review-ului

7. Recomandări și feedback:

- Oferiți feedback specific și constructiv pentru a îmbunătăți codul.
- Identificați și sugerați soluții la problemele găsite.
- Evidențiați lucrurile bune din cod.

8. Rezumat și acceptare:

- O scurtă concluzie a review-ului și o decizie cu privire la acceptarea sau respingerea codului.