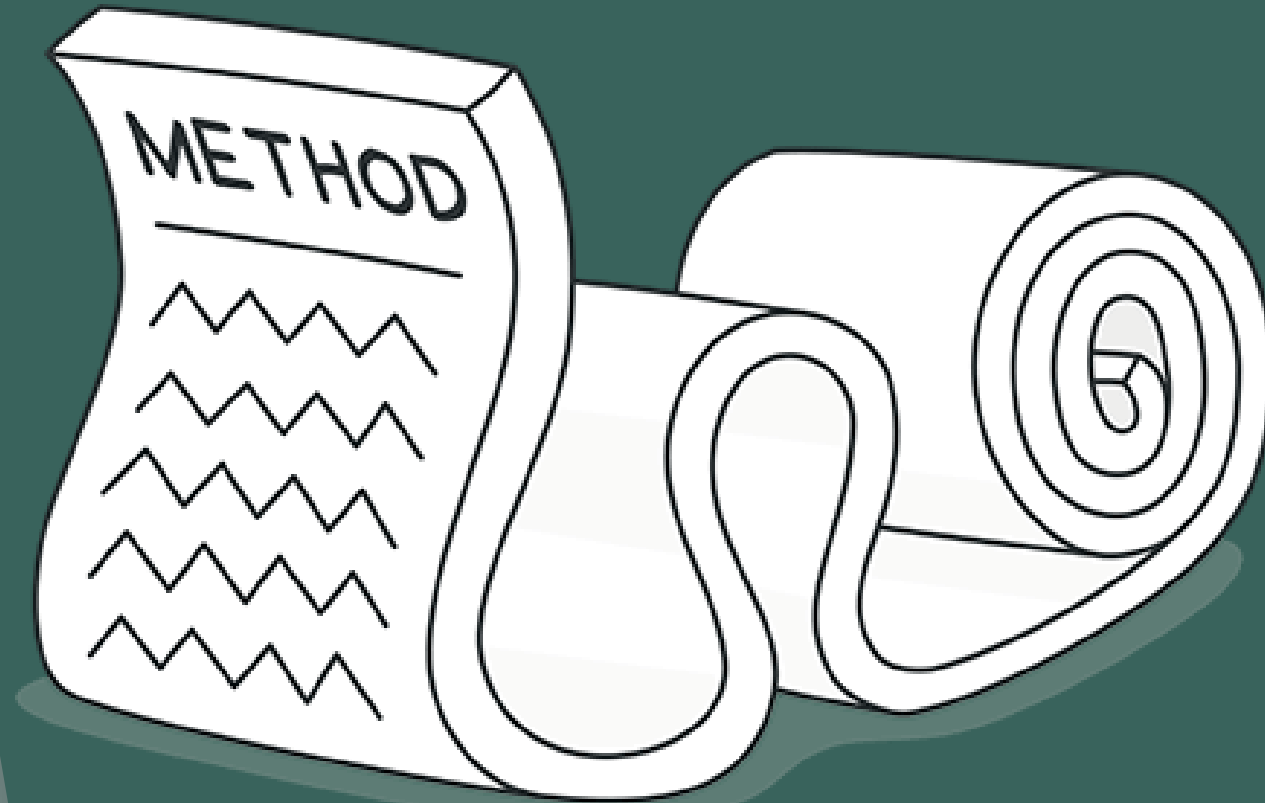


# Uvod u softversko inženjerstvo

Čiste funkcije

Nikola Luburić  
[nikola.luburic@uns.ac.rs](mailto:nikola.luburic@uns.ac.rs)



Refaktorisiranje pozorišta

Aspekti čistih funkcija

Čiste funkcije

GPT i čiste funkcije

## ✓ 8. Konstruiši čiste funkcije

✓ 1. Segmentiraj dugačku funkciju spram regiona povezane logike

✓ 5. Umanji složenost funkcije

✓ 2. Reorganizuj logiku funkcije radi smanjenja njene složenosti

✓ 3. Izdvoj složenu logiku u posebnu funkciju

✓ 4. Imenuj rezultat složene logike

✓ 6. Smanji broj parametra funkcije upotrebom prikladne strategije

✓ 7. Definiši jedan zadatak koji funkcija rešava

*Šta je glavni  
izazov?*

*Organizovati  
kohezivne  
regione*

✓ 8. Konstruiši čiste funkcije

✓ 1. Segmentiraj dugačku funkciju spram regiona povezane logike

✓ 5. Umanji složenost funkcije

✓ 2. Reorganizuj logiku funkcije radi smanjenja njene složenosti

✓ 3. Izdvoj složenu logiku u posebnu funkciju

✓ 4. Imenuj rezultat složene logike

✓ 6. Smanji broj parametra funkcije upotrebom prikladne strategije

✓ 7. Definiši jedan zadatak koji funkcija rešava

# Izazov

**Cilj:** Analiziraj kod i definiši regione povezane logike koje bismo izdvojili u posebne funkcije

1. Prođi kod i identifikuj kohezivne regione logike
2. Za svaku funkciju koju bismo izdvojili definiši adekvatan naziv
3. Analiziraj šta bismo još mogli da uradimo za **redukciju složenosti**

Diskutujte u grupicama od 2-3

Ne mogu da vas kontrolišem, morate sami

Posle 10 minuta nastavljamo, brzo smirivanje

<http://tiny.cc/SIMS-FR>

✓ 8. Konstruiši čiste funkcije

✓ 1. Segmentiraj dugačku funkciju spram regiona povezane logike

✓ 5. Umanji složenost funkcije

✓ 2. Reorganizuj logiku funkcije radi smanjenja njene složenosti

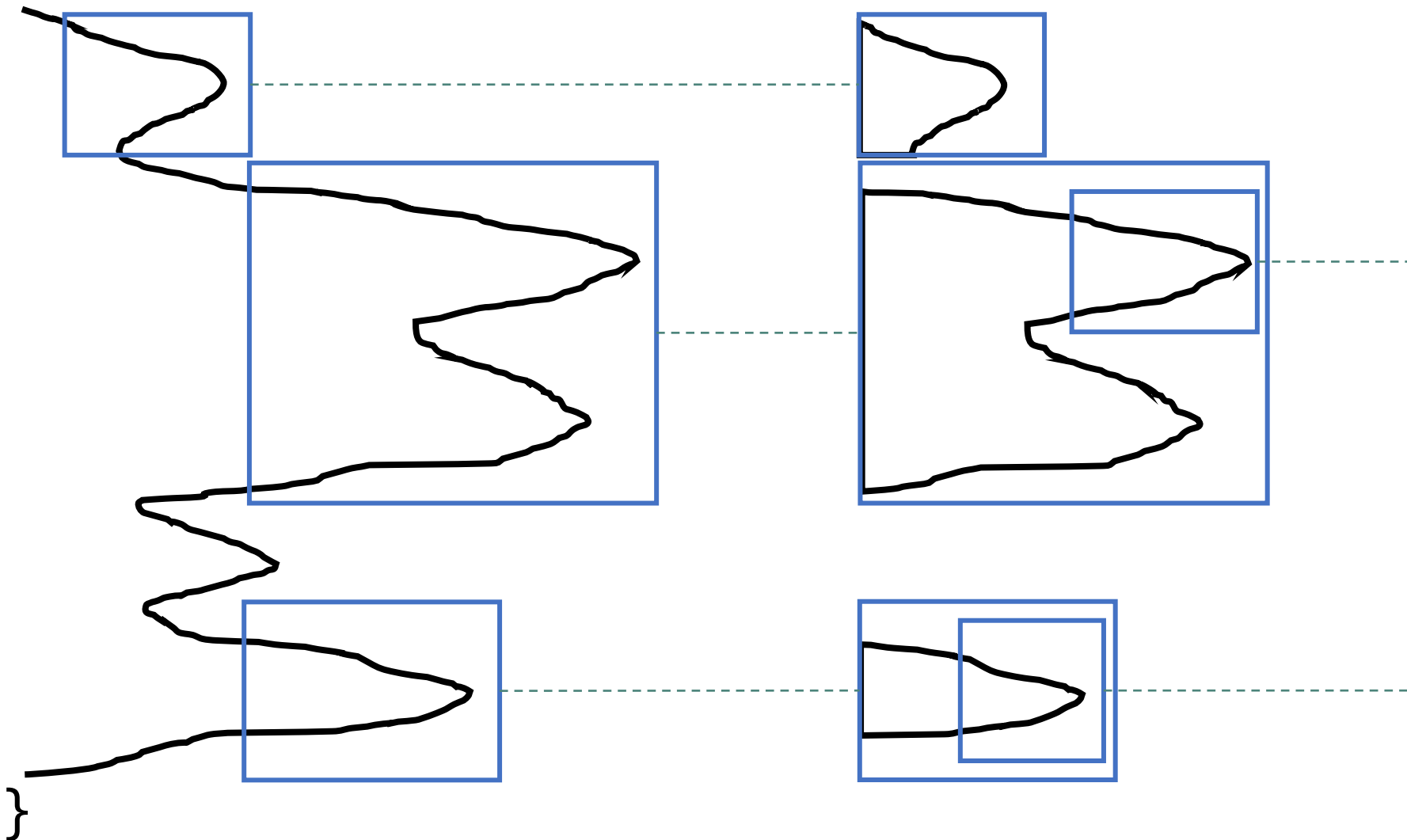
✓ 3. Izdvoj složenu logiku u posebnu funkciju

✓ 4. Imenuj rezultat složene logike

✓ 6. Smanji broj parametra funkcije upotrebom prikladne strategije

✓ 7. Definiši jedan zadatak koji funkcija rešava

```
importModel(...) {
```



✓ 8. Konstruiši čiste funkcije

✓ 1. Segmentiraj dugačku funkciju spram regiona povezane logike

✓ 5. Umanji složenost funkcije

✓ 2. Reorganizuj logiku funkcije radi smanjenja njene složenosti

✓ 3. Izdvoj složenu logiku u posebnu funkciju

✓ 4. Imenuj rezultat složene logike

✓ 6. Smanji broj parametra funkcije upotrebom prikladne strategije

✓ 7. Definiši jedan zadatak koji funkcija rešava



✓ 8. Konstruiši čiste funkcije

✓ 1. Segmentiraj dugačku funkciju spram regiona povezane logike

✓ 5. Umanji složenost funkcije

✓ 2. Reorganizuj logiku funkcije radi smanjenja njene složenosti

✓ 3. Izdvoj složenu logiku u posebnu funkciju

✓ 4. Imenuj rezultat složene logike

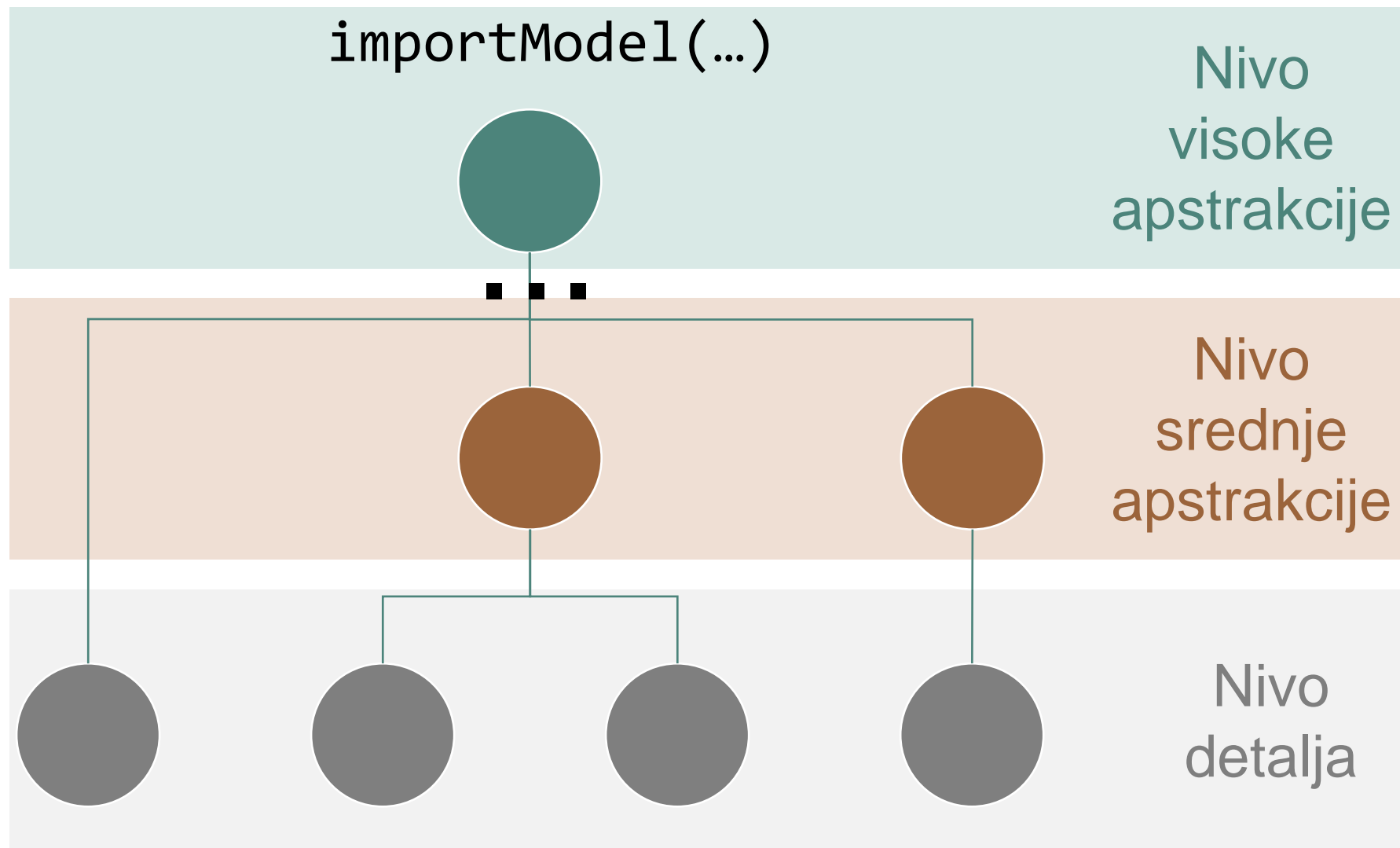
✓ 6. Smanji broj parametra funkcije upotrebom prikladne strategije

✓ 7. Definiši jedan zadatak koji funkcija rešava

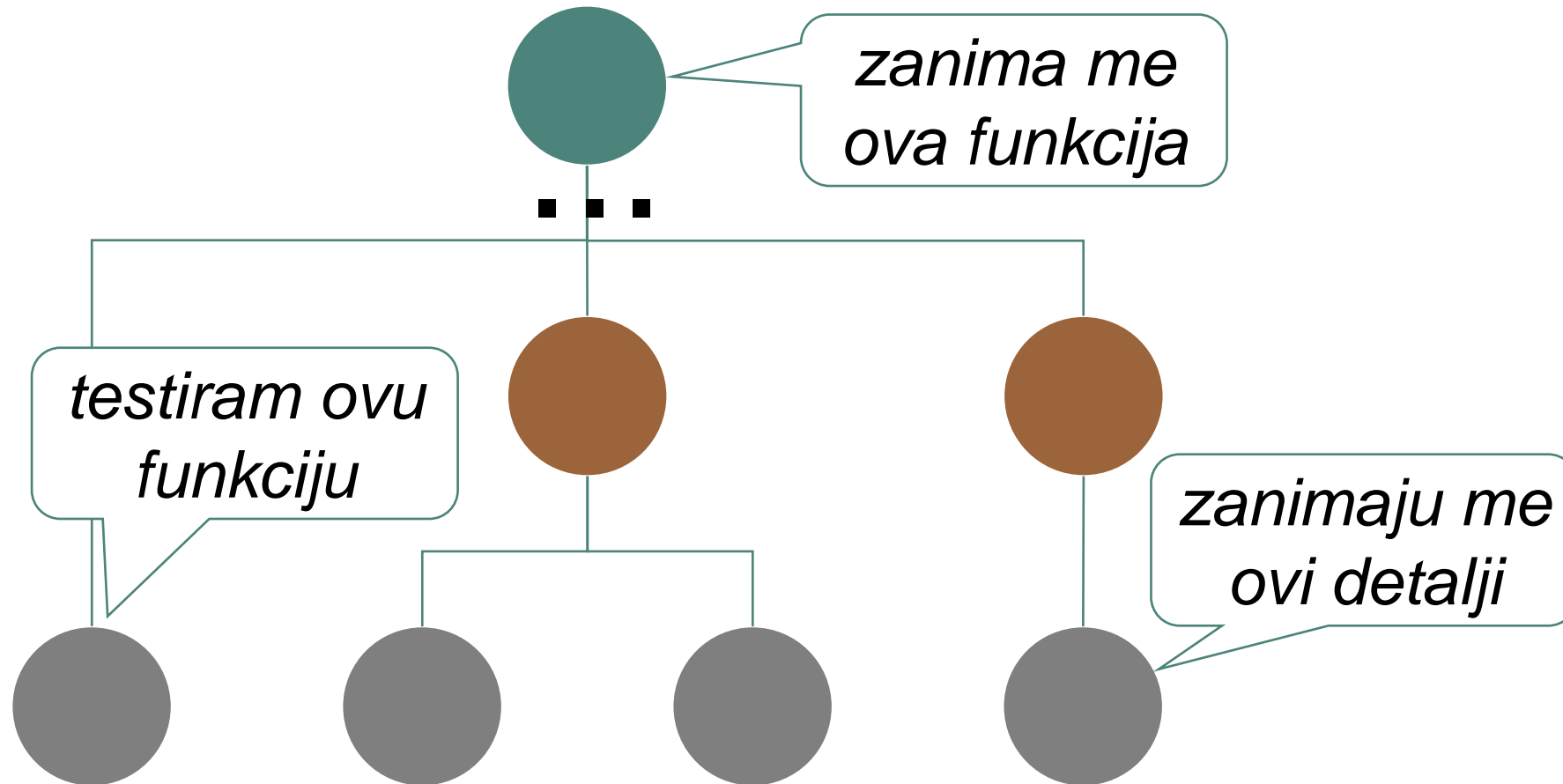


Čista funkcija sadrži:

1. Jasno definisan zadatak koji rešava i koji je rezimiran nazivom funkcije.
2. Ograničenu količinu složene logike, bez mnogo ugnježdavanja i složenih izraza.
3. Semantički povezan skup naredbi koje staju u mali broj linija koda.
4. Mali broj parametra, izuzev funkcija čiji zadatak je da kreira složeni objekat (npr. konstruktor).



```
importModel(...)
```



# Uticaj čistih funkcija na održivost?

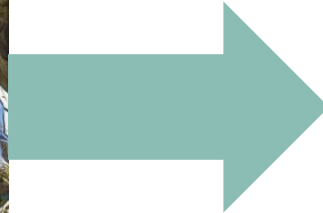
ISO 25010 razlaže održivost na:

## Maintainability

- Modularity
- Reusability
- Analysability
- Modifiability
- Testability

- **Modularnost** - Stepen u kom je program sastavljen iz diskretnih modula tako da promena modula ima minimalan uticaj na ostale
- **Ponovna iskoristivost** - Mogućnost modula programa da se koriste na više mesta unutar sistema ili u više od jednog sistema
- **Analizabilnost** - Jednostavnost analize modula radi razumevanja rada, dijagnostikovanja uzroka otkaza ili planiranja izmene
- **Promenljivost** - Jednostavnost izmene modula bez uvođenja defekta ili umanjenja kvaliteta
- **Testabilnost** - Lakoća definisanja testnih kriterijuma za modul i provera njihove ispunjenosti







**You**

Write a python script that loads a jpg file and processes it to discover all black hexagon shapes on the image. For each discovered hexagon, the script should save a new image that contains the content of the black hexagon

<https://chat.openai.com/share/eda80245-6029-42e1-b5b2-9aca7348f388>

1 sat kasnije...

