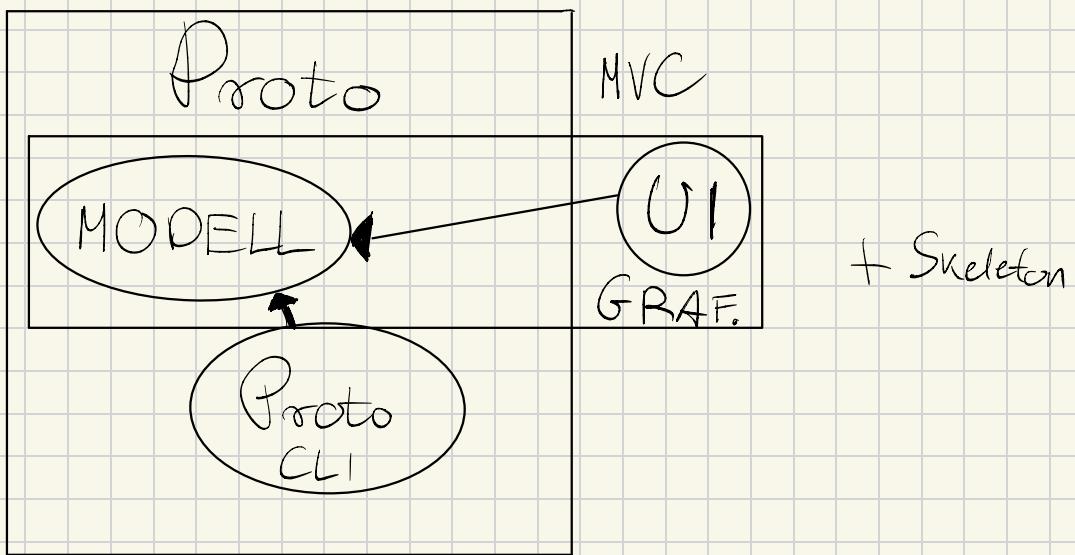


Projekt



- Sakk Java kód

Felépítés:



ProtoCLI-nak
előre megírt
teszteseteket
adunk...

- Esemény vezetett strukturált modellt osztunk (üzleti)
- Módosítások fognak kiijenni, amiket sikeresen meg kell oldanunk, tehát erre felkészülve kell megírni a programot
- Együtt Kódolás Preferred
- Git Hub használata
- %-ok egész felévre érvényesek nem egy adott szakaszra
- mindenki csinálja valamit (eset bekéndezi és mindenki fájul)
 - kedvelt szerepek kiismerősek (mindehelyi szemesse, amit csinál => nem szabadunk meg)

- minden szükségből meg kell lennie 41%-nak
- DEFO -nál faktoria kell lemutatásban
- ~~Gingen~~ - TLG
by:

Testeset:

Input:

...
~~~  
~~~

elvárt kimenetnél ne fix print, mert úgy se azt fogja adni
és minden feliratnál fog → ~~F~~

Első leadásra:

- Feladat leírás:
 - nagy alkotói szabadság, de feleljen meg a leírásnak
 - Ne írjunk bele objektumot amit nem fogunk tudni megvalósítani
Vagy nincs még rögzítve a leírásban
- Ne legyen túlragonva
- Alap USE-CASEK, más ne megy (nyomogatja a gombot mit csinál)
 - Különleges nem USE-Case ✗
 - Játék elindítása USE-CASE ✓
 - Le van írva minden USE-CASEK vannak
 - Óra utolsó USE-CASE

Feladat = Sötér + USE-CASE leírásaihoz

Jelenléti leadás

Objektum orientált szöveg

<u>„class”</u>	<u>„method”</u>	<u>állapot/tulajd. / id</u> előző köröktől lépés - változás
Főnél	lge	JFn 1 Jlge de füntes
Miskolci	Maverick	Vékony_Md!
Város(püjsg)	Közlekedés	Vastag_Mb
Havazás	Letapos_Hd	Sérhatállan_Súv
Auto	lefagy	legrövidítés_út
Hd!	Koccancs	
Jégpáncél	takarít_át	
Súv		
Lakás		
Munkahely		

Pl: útnak esy járatokat

tulajdonság \Rightarrow ha van kocc,
csak azt a tulajt
hely módosítani
és már is megvagyunk
(nem kell törlni utat)

List <T>

Set <T>

Map <K, V>

Út végigjárás:

Graf eszik pontjából teleport másik pontjába és ezt grafikusan lecserítjük

-Design Patterns: codinghorror

refactoring.guru

| itt található Rögy mit ne csinálunk

Coupling

Pl: ha bejönnek a buszok és sok új helyre fel kell venni

Code Sonell:

nézzünk rá! Shotgun Surgery