

Sabloane de proiectare: Observer

Cosmin Varlan

VESTEA BUNA:

PE 22 VINE ADOBE :D

Sabloane de proiectare (design patterns)

- Ne usureaza viata....
- Ne ajuta sa organizam mai bine treaba
- Putem sa intelegem “mai bine” OOP
- Uneori pot reduce oridunl de complexitate al problemei....
- ... sau fac definitiile obiectelor mai usor de inteles...

Sabloane de proiectare

Sabloane...	Sabloane...	Sabloane...
Strategy	Adapter	Proxy
Decorator	Facade	Command
Factory	Template	Mediator
Observer	Builder	Model-View-Controller
Chain of responsibility	Iterator	Recycle Bin
Singleton	Composite	The Circular
Fly Weight	State	Bridge

... si multe altele....

Recomandare:



Da, o gasiti si pe torente...

Sabloane de proiectare

Sabloane...	Sabloane...	Sabloane...
Strategy	Adapter	Proxy
Decorator	Facade	Command
Factory	Template	Mediator
Observer	Builder	Model-View-Controller
Chain of responsibility	Iterator	Recycle Bin
Singleton	Composite	The Circular
Fly Weight	State	Bridge



Despre asta vom vorbi azi...

Problema:

- Ionel, Maria, Gigel si Ioana sunt foarte interesati de ActionScript. Ei vor sa afle cand va apare noua versiune a flexului, cea care incorporeaza noile facilitati ale Flash Playerului 10 (3D, engine nou pentru formatarea textului, pixelBenders si multe altele).

Problema:

- Ca sa afle, ei intra zilnic pe situl Adobe, chiar de doua ori pe zi sau in cazul loanei care e super-maniaca de AS3 de trei ori.
- Adobe s-a saturat sa faca trafic mereu si mereu spre aceleasi persoane si din cauza asta vine cu o solutie:....

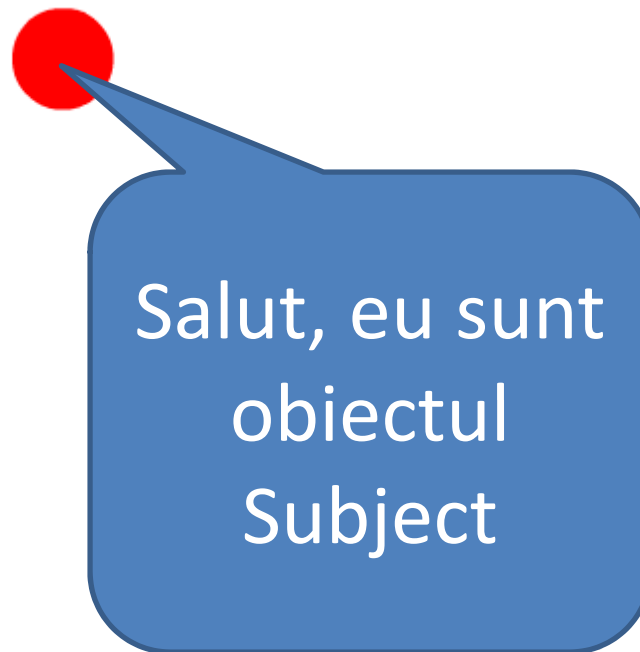
Problema - solutia:

- Hai ca va zic eu cand apare flexul ala, dati-mi voi adresele voastre de mail si cand o apare va trimiti cate un mail la fiecare.

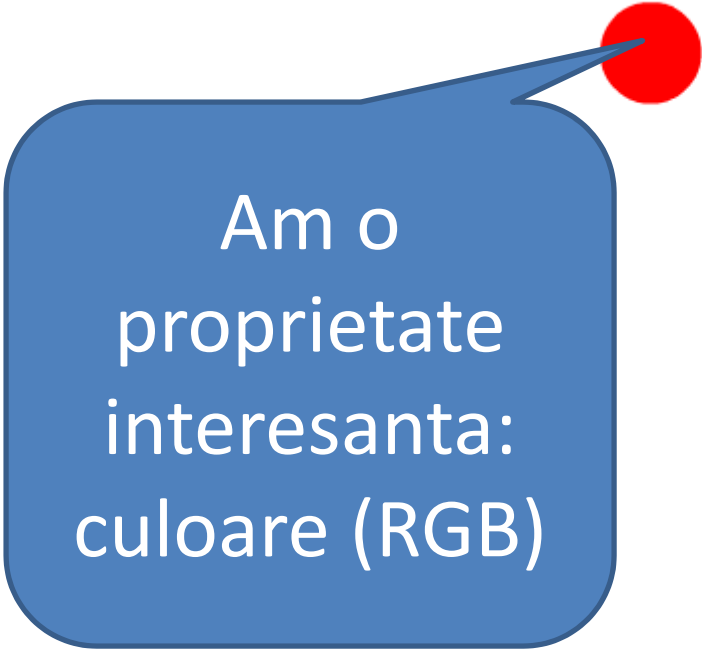
Observer...

- In exemplul dat, Ionel, Maria, Gigel si Ioana (nu neaparat in ordinea asta) sunt observatori. Ei se “inregistreaza” intr-un fel la entitatea pe care vor sa o interogheze (si pentru care vor sa afle cand i s-a modificat o anumita proprietate) .
- Obiectul observat se numeste “Subject”
- Obiectele care observa se numesc “Observers”

Pasii “de urmat” pentru a deveni un observer:



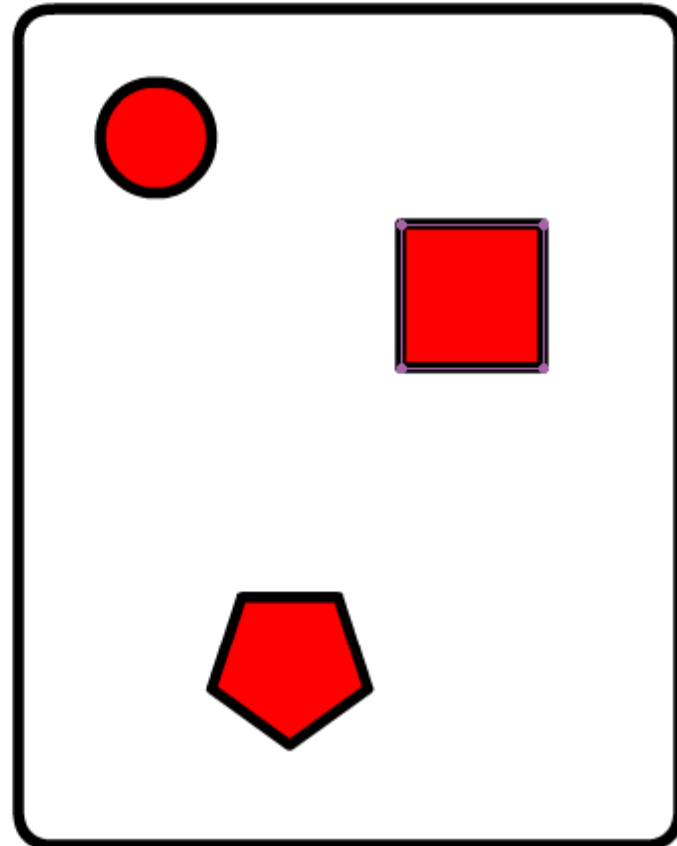
Pasii “de urmat” pentru a deveni un observer:



Am o
proprietate
interesanta:
culoare (RGB)

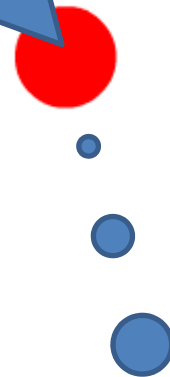
Pasii “de urmat” pentru a deveni un observer:

Obiectele din dreapta sunt cele interesate de culoarea mea

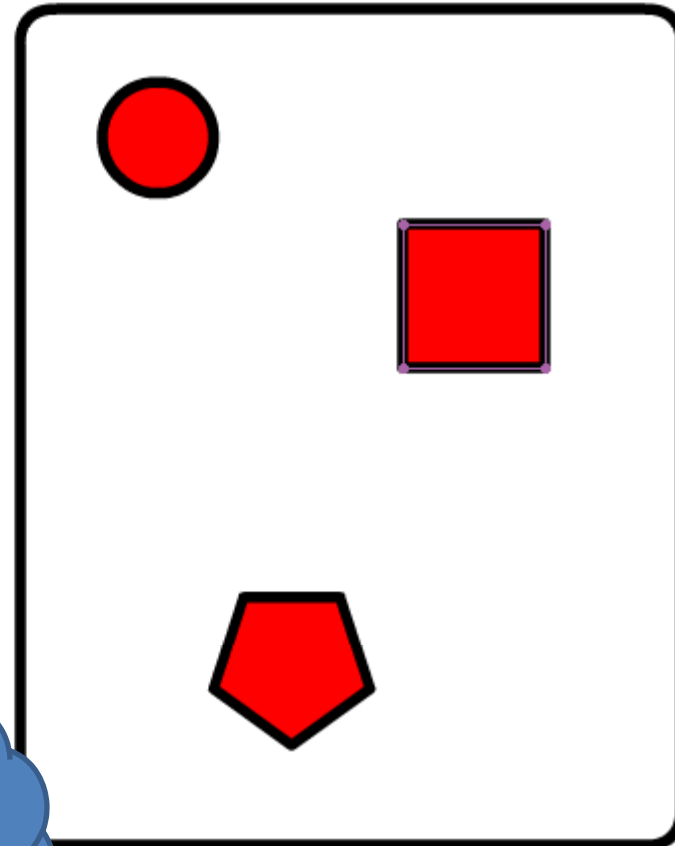


Pasii “de urmat” pentru a deveni un observer:

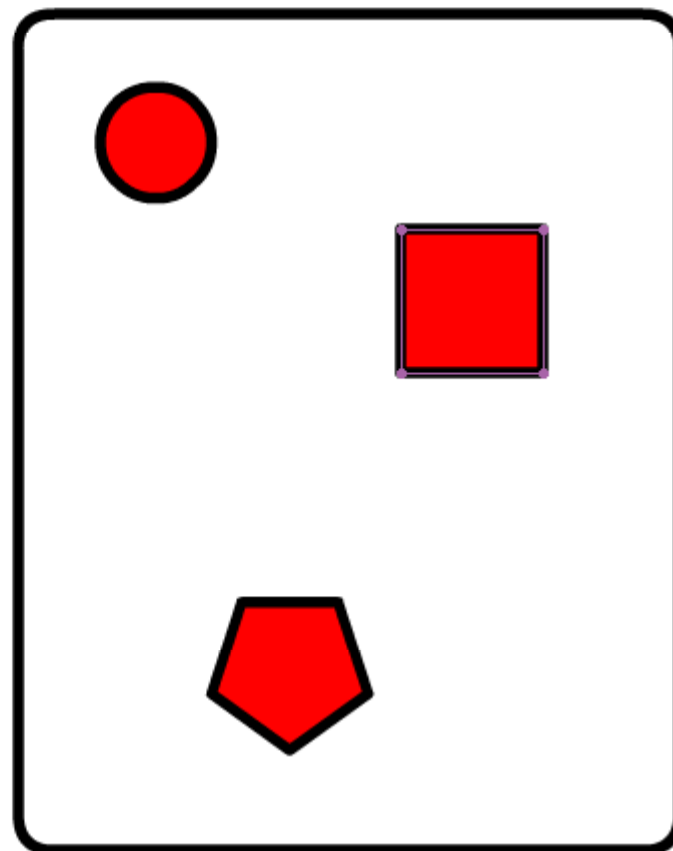
Atunci cand
imi schimb
culoarea, le
transmit si ei
isi vor
schimba
culoarea
pentru a fi ca
mine



Because I'm
cool

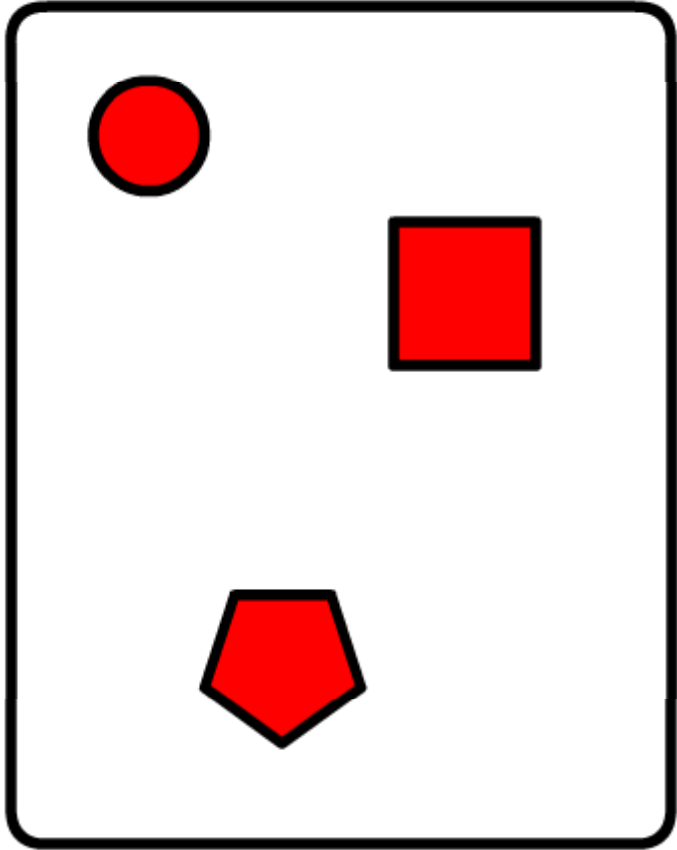


Pasii “de urmat” pentru a deveni un observer:



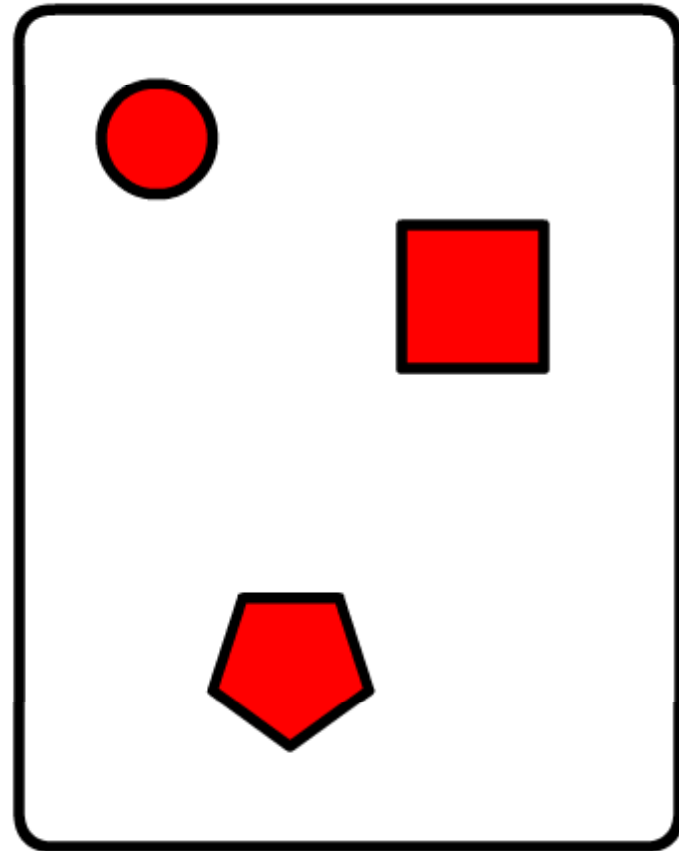
Pasii “de urmat” pentru a deveni un observer:

Am o proprietate de tip Array in care pastrez referinte catre observatorii mei...

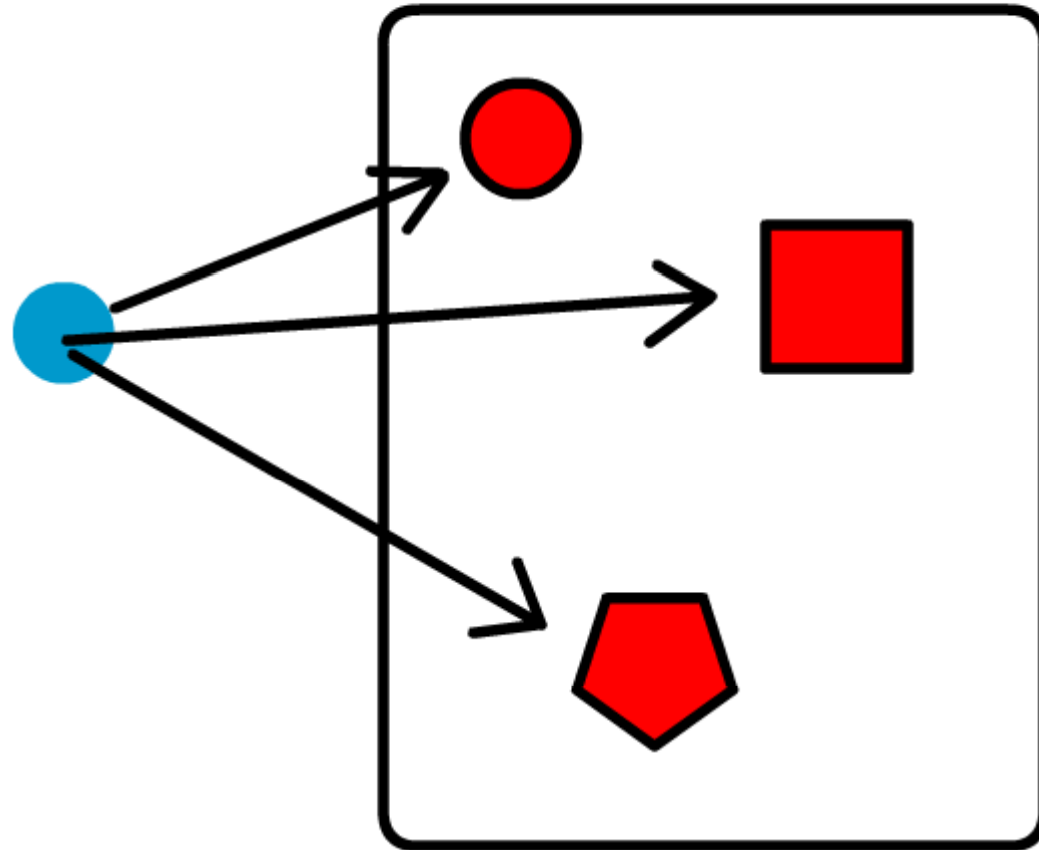


Pasii “de urmat” pentru a deveni un observer:

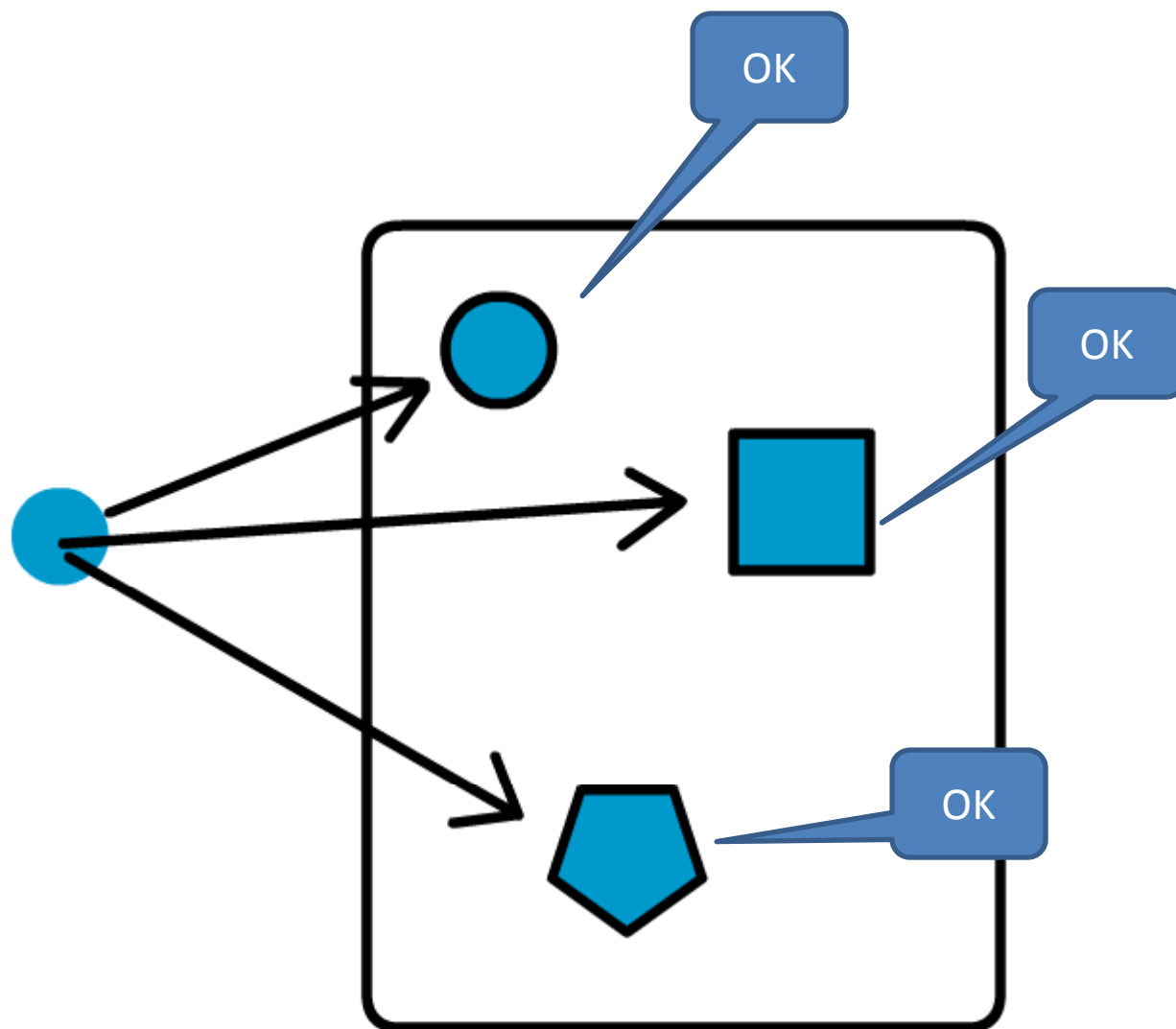
Pentru fiecare
obiect din
Array ii voi
transmite
noua culoare.



Pasii “de urmat” pentru a deveni un observer:

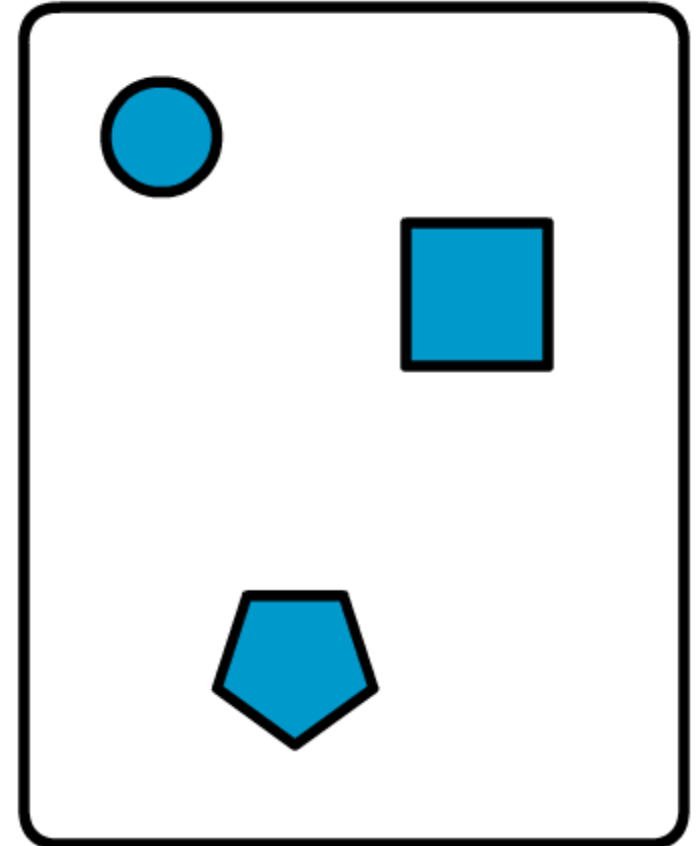


Pasii “de urmat” pentru a deveni un observer:




Pasii “de urmat” pentru a deveni un observer:

Momentan Array-ul
cu referinte este:
Observatori = {Cerc,
Patrat, Pentagon}

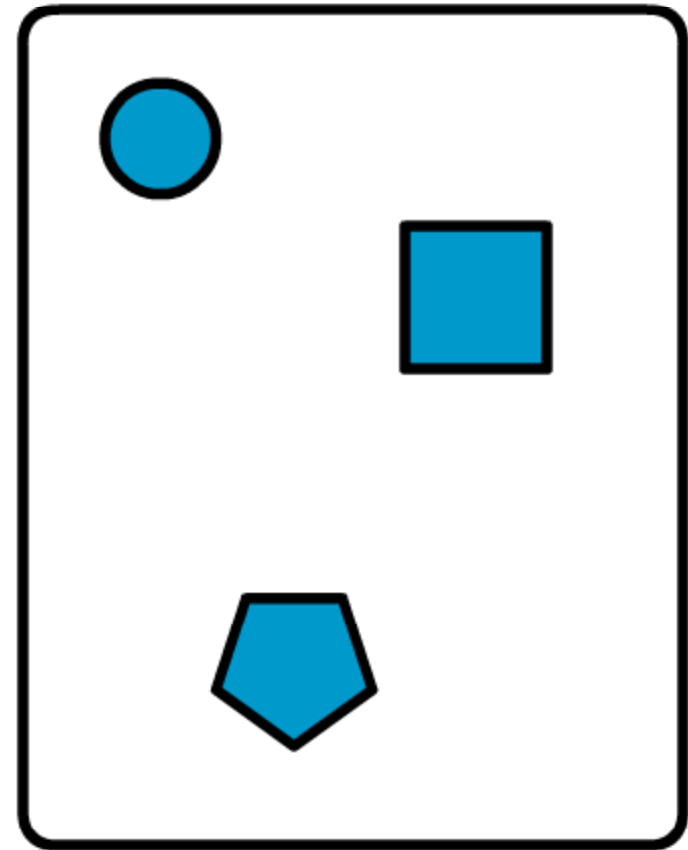


Pasii “de urmat” pentru a deveni un observer:

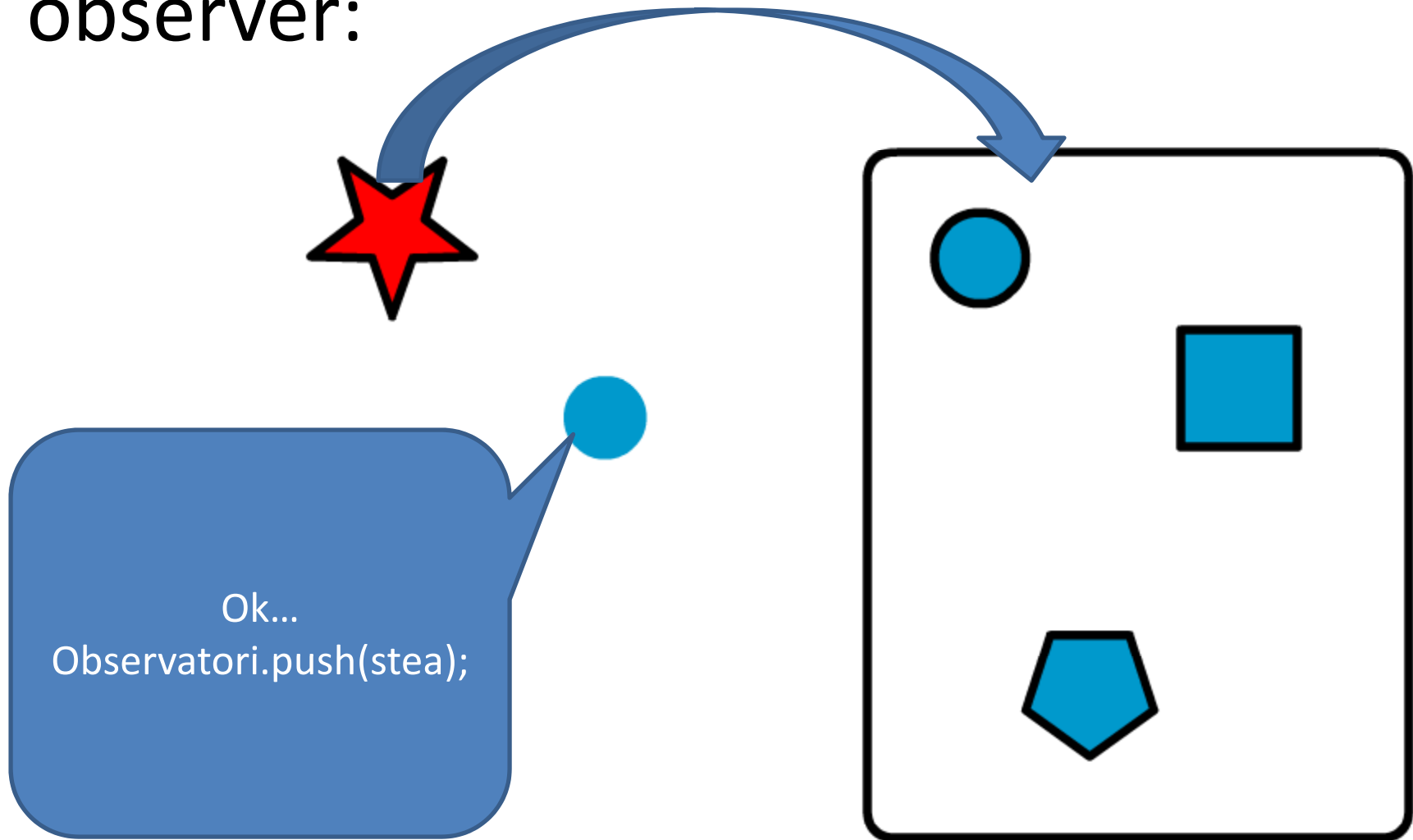


Vreau si eu sa fiu Cool
(ca tine).

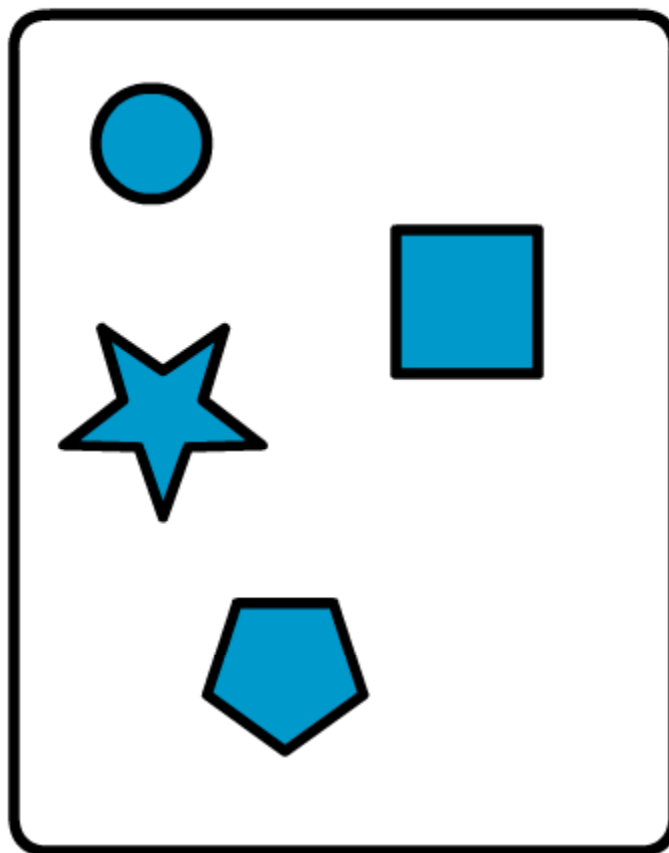
```
Cerc.register("me")...
```



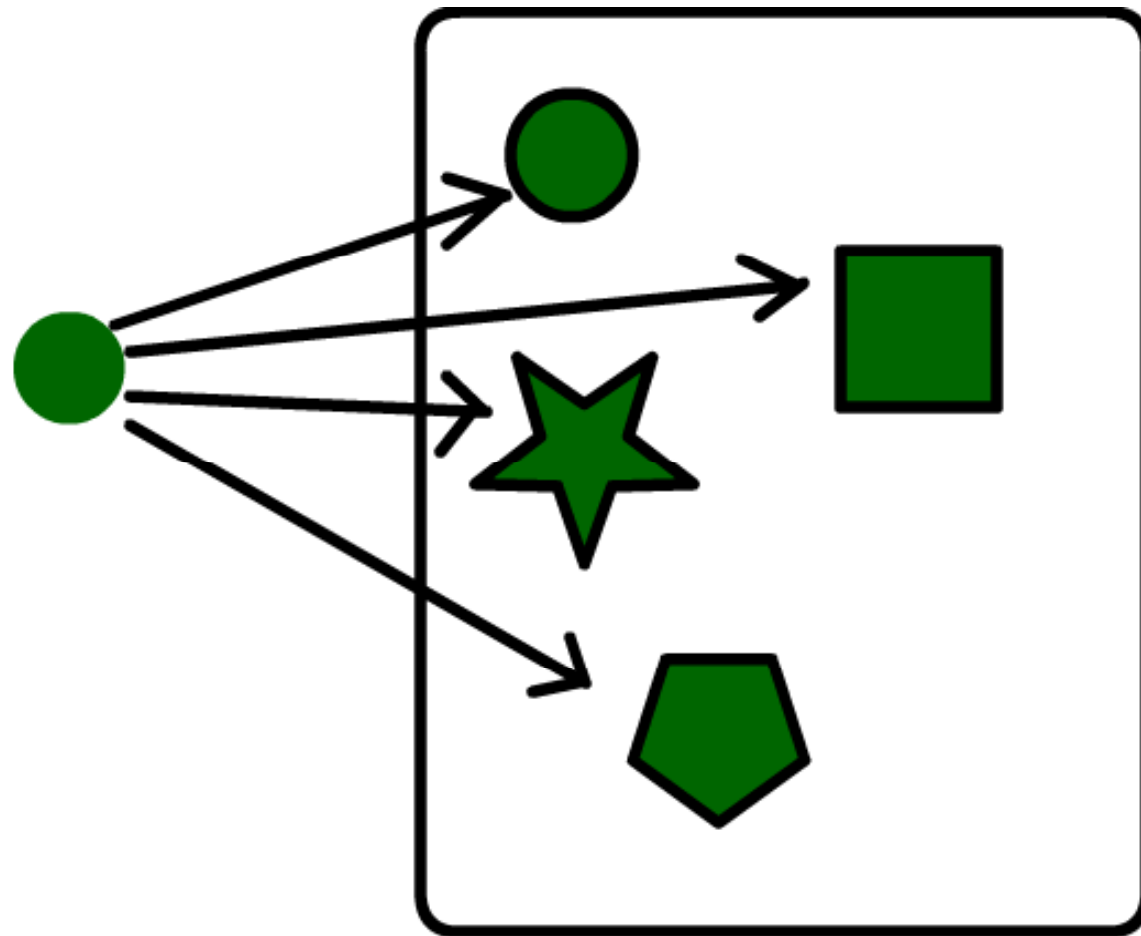
Pasii “de urmat” pentru a deveni un observer:



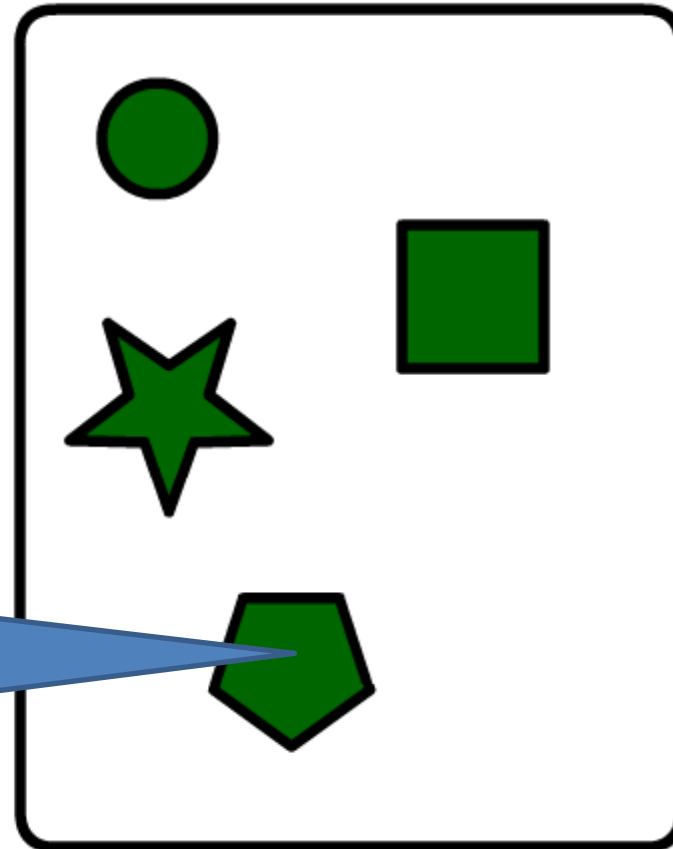
Pasii “de urmat” pentru a deveni un observer:



Pasii “de urmat” pentru a deveni un observer:

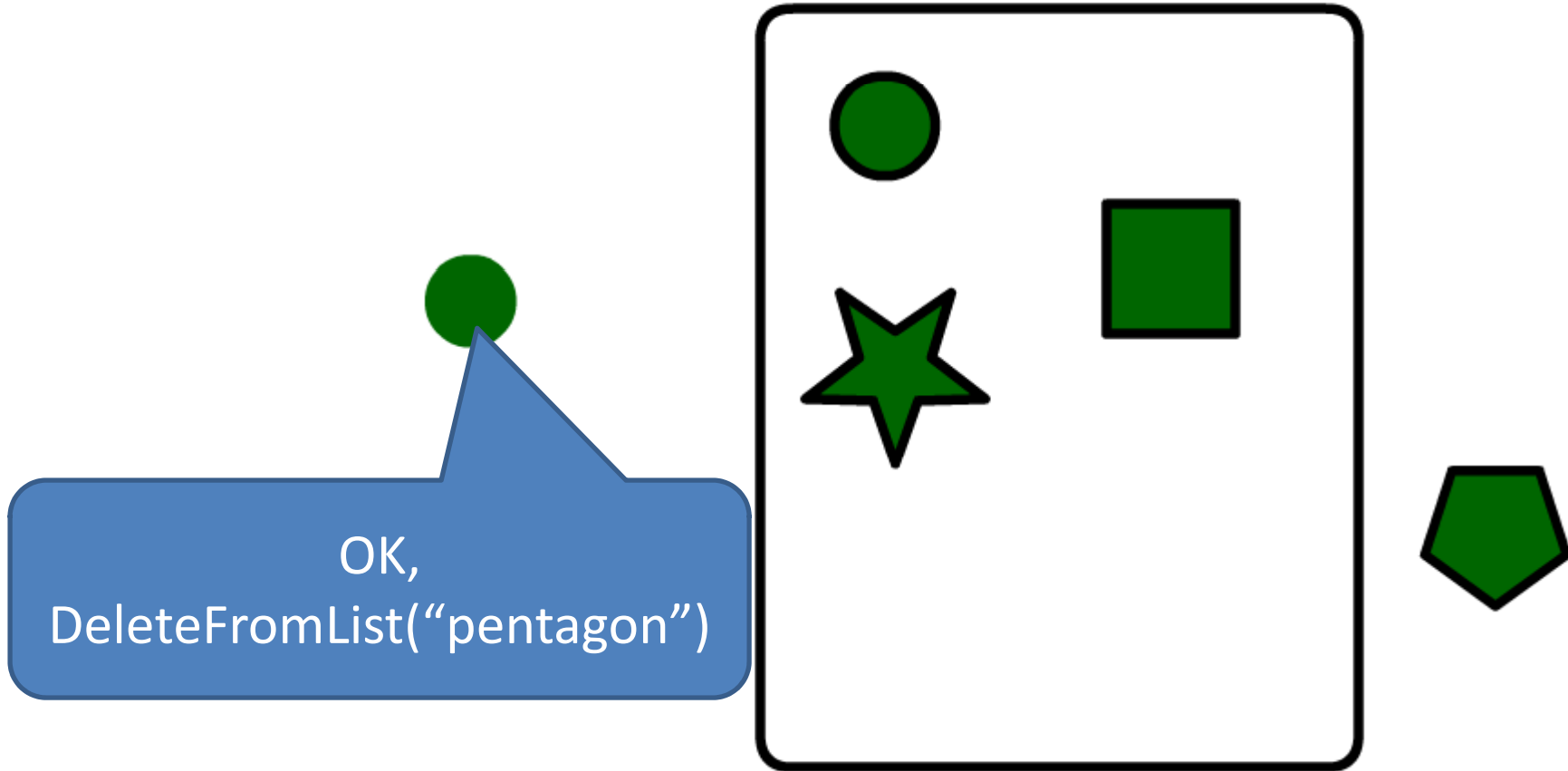


Pasii “de urmat” pentru a deveni un observer:

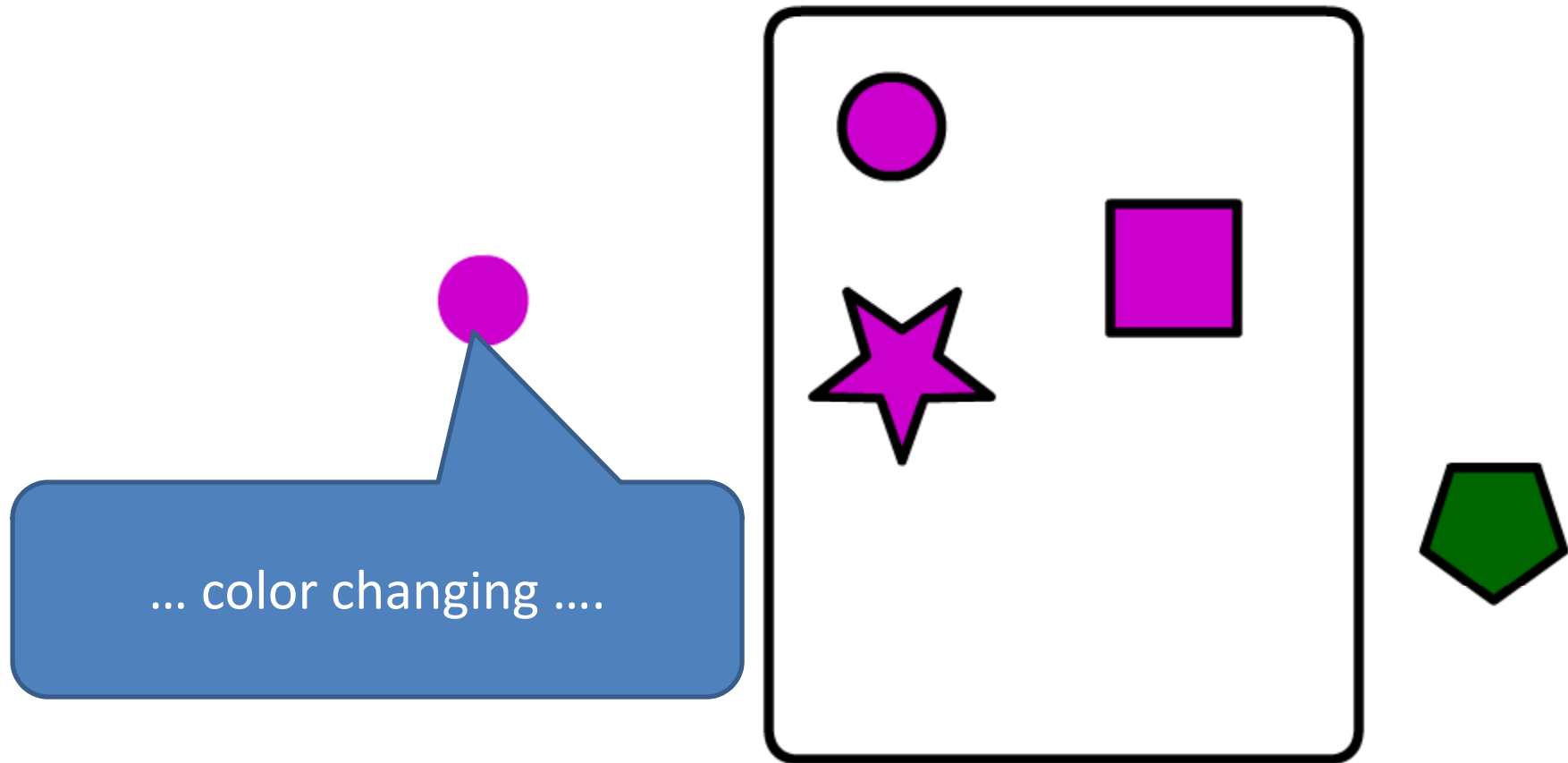


OK, ai devenit prea mandru, nu te mai plac,
`Cerc.unregister("me")`

Pasii “de urmat” pentru a deveni un observer:



Pasii “de urmat” pentru a deveni un observer:



Cum se realizeaza ?

- Pentru a putea inregistra / elimina observeri, obiectul Subject trebuie sa posede cate o functie ce va fi apelata de observatori (posibili sau existenti).
- In obiectul-subiect trebuie sa existe un Array ce va contine referinte catre obiectele inregistrate.
- register va adauga o referinta,
- unregister va elimina o referinta

Cum se realizeaza ?

- Daca obiectele - observatori sunt heterogene, ar fi bine sa implementeze o interfata : sa primeasca toti modificarea prin aceeași functie.. “receiveData”

Subiectul:

- Interfata cu:
 - registerObject()
 - unregisterObject()
 - notifyObservers()

Eventual (vedem acusi de ce):

- getState();
- setState

Observerul:

- Interfata cu:
 - update

O alta idee... (Java - like)

- De ce sa trimite fortat datele catre observatori?
- Am putea mai degraba doar sa informam observatorii ca o proprietate s-a schimbat si sa facem acea proprietate disponibila printr-un getter. Daca Observerul este interesat, va prelua singur informatia (interesant cand sunt multe proprietati ce pot fi modificate)
- De asta aveam nevoie de getState / setState de mai sus.


Problema:

- Avem un numar mare de buline (400) care se “plimba” intr-un plan.
- Cu cat o bulina are mai multi vecini cu atat va fi mai mare (schimbam diametrul in functie de distanta: de exemplu la diametrul bulinei curente se adauga $1/(\text{distanta} * 10)$ dintre ea si toate celelalte buline).
- De la o anumita valoare incolo nu mai conteaza (de exemplu daca distanta este mai mare ca 20, diametrul se va mari cu $1/200$ - insignifiant)

Problema:

- Ar fi ok daca am imparti planul 2D in patrate cu latura de 20px si de fiecare data cand un punct intra intr-un patrat, se inregistreaza acelui patrat.
- Cat timp este in patrat primeste notificari de la acesta cu lista obiectelor care se mai afla in patrat
- In functie de aceasta lista, va reduce numarul calculelor de la 400 la numarul pnctelor din patrat.

Problema:

- Atunci cand iese din patrat, face unregister si face register la noul patrat (desi aici ar fi bine sa faca unregister/register) din doua in doua patrate (adica sa se inregistreze in patratul curent si in patratele adiacente).
- Motivul este ca daca trece dintr-un patrat aglomerat intr-unul gol se stinge prea brusc.
- Se asteapta implementatori 

NU UITATI !!!

PE 22 VINE ADOBE :D

Fiti prezenti...😊...