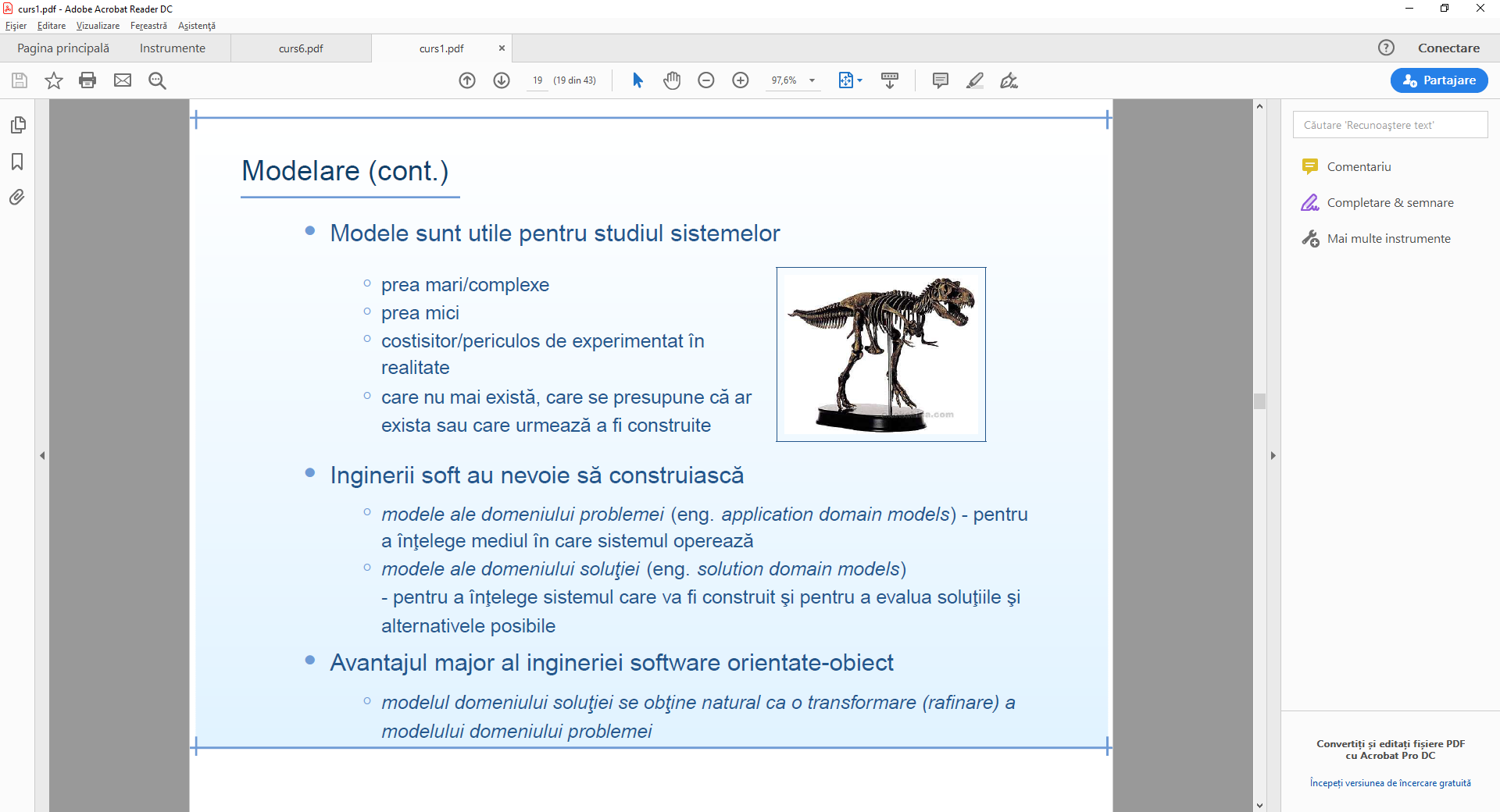
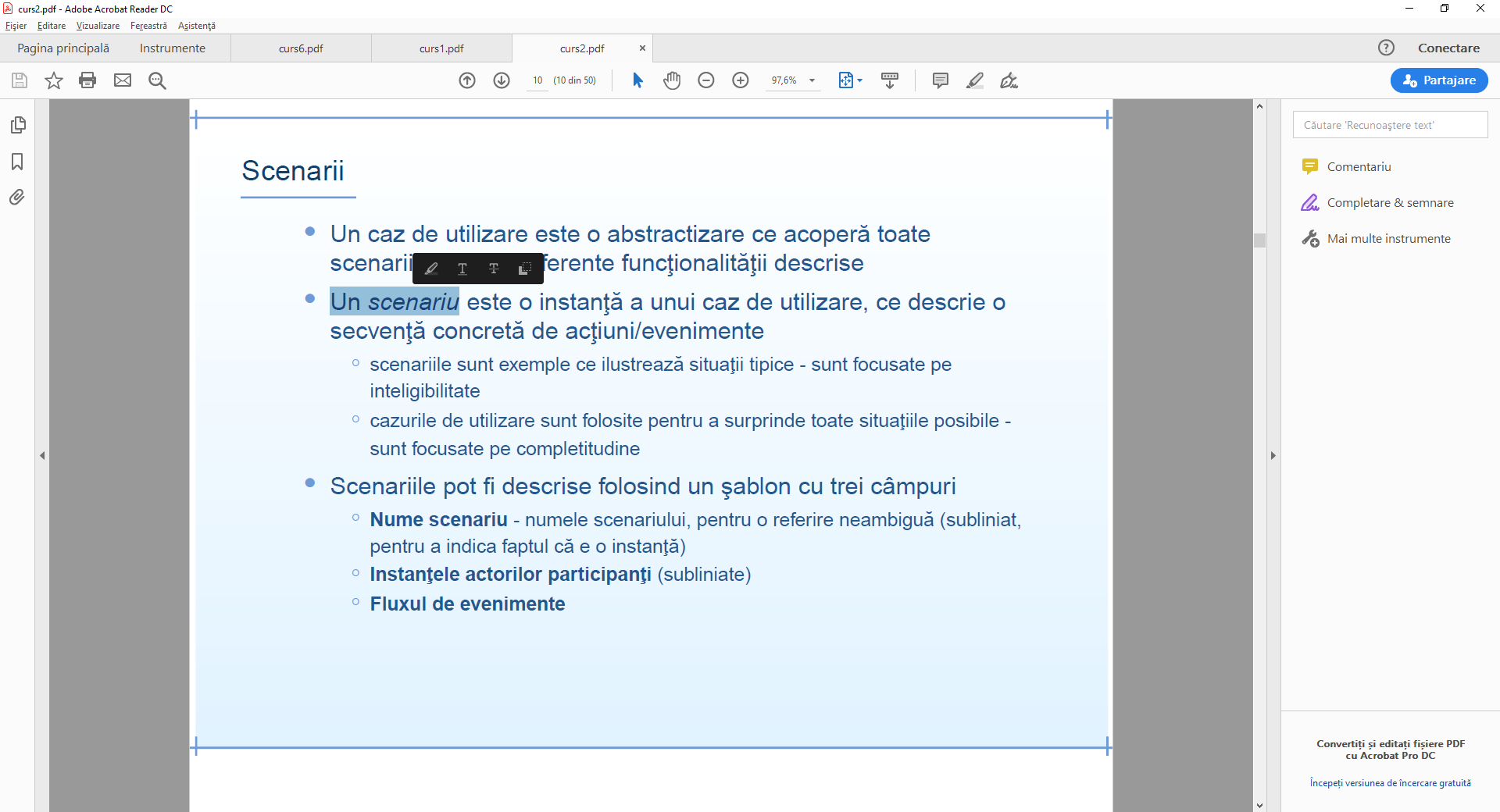
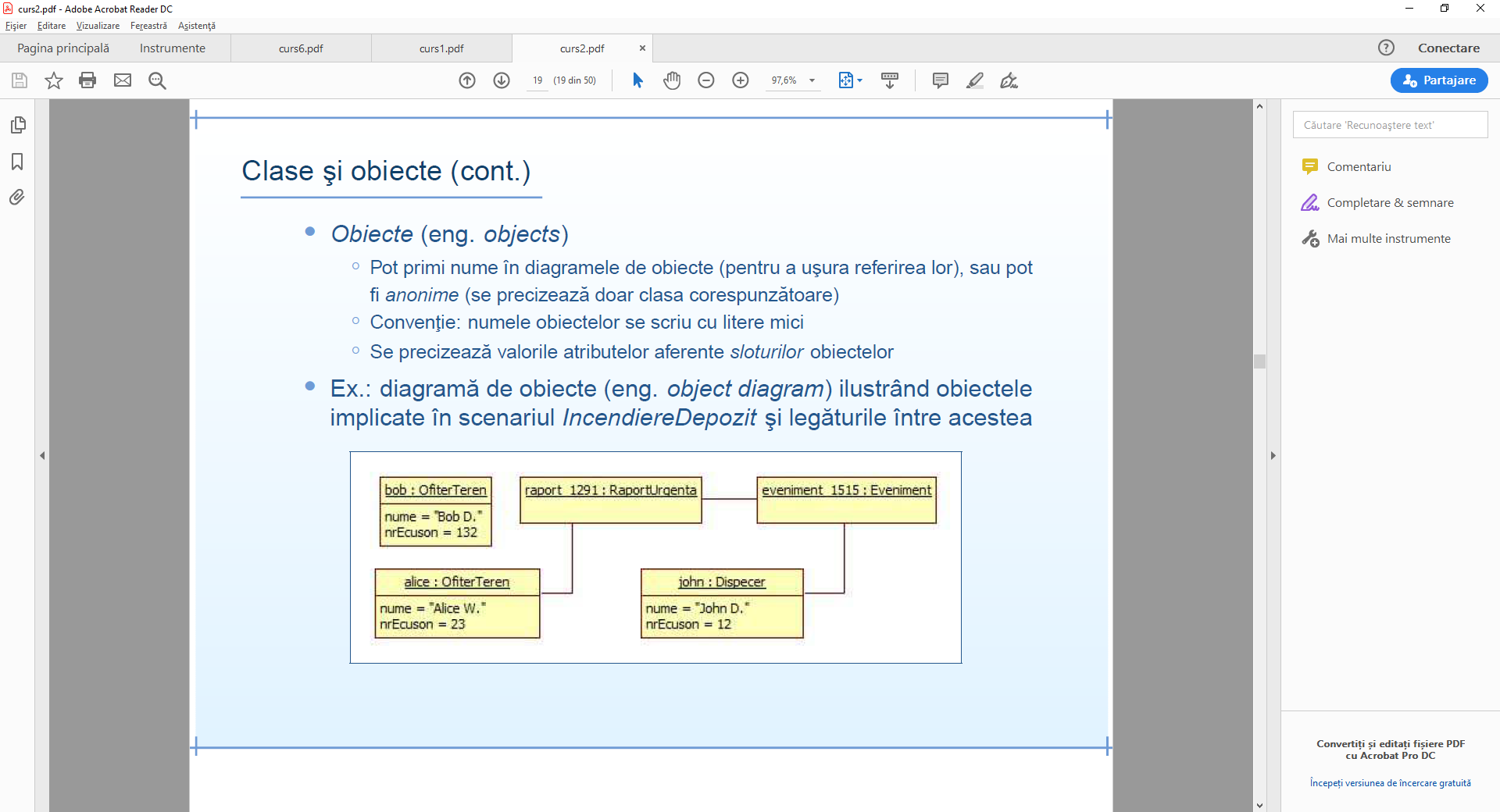
A (40%). Raspundeti succint la urmatoarele cerinte/intrebari

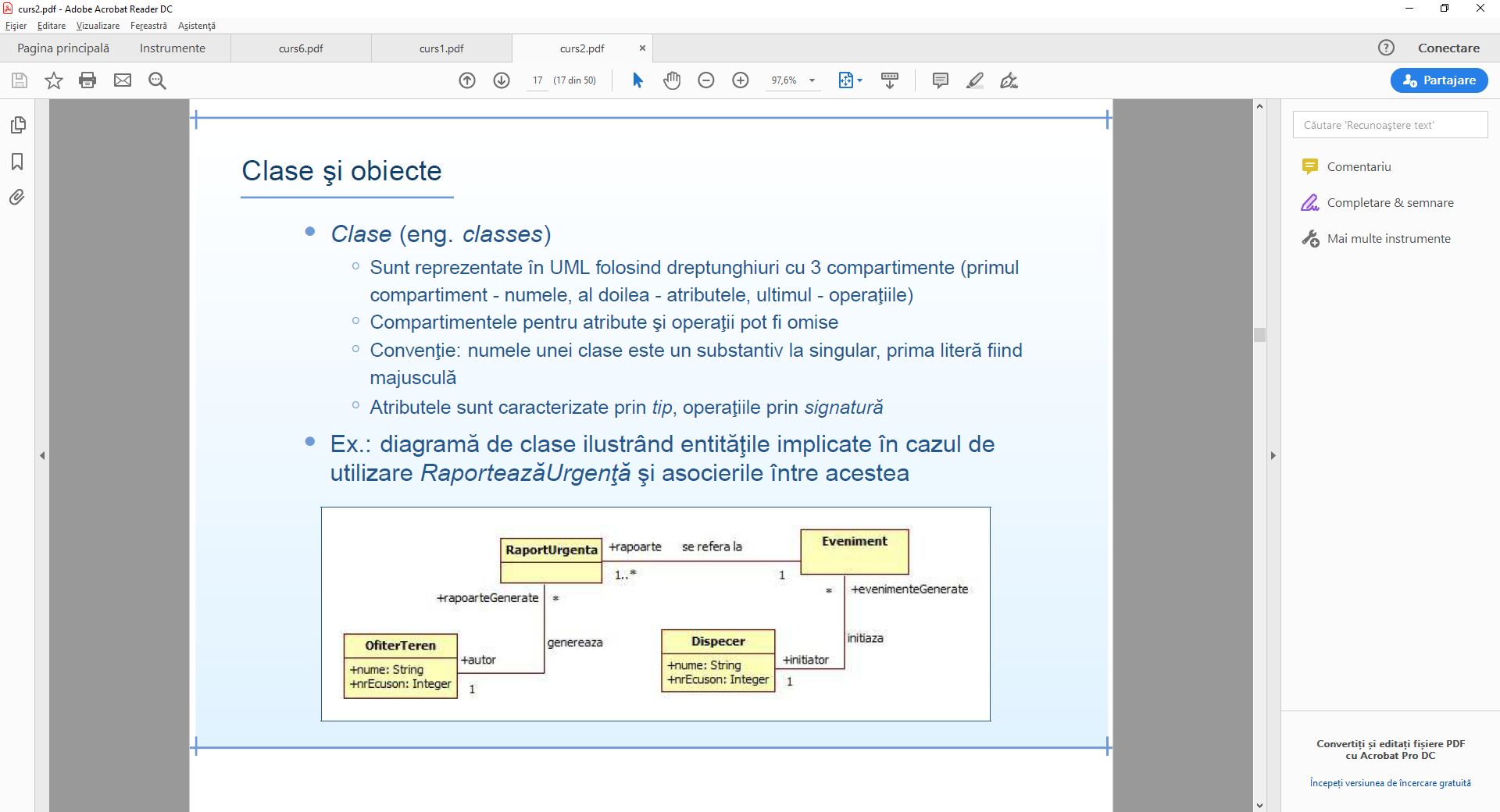
A1. Explicati diferenta dintre domeniul problemei si domeniul solutiei. (1.5p)

A2. Definiti conceptul de scenariu. (1.5p)

A3. Dati un exemplu de diagrama de obiecte corespunzatoare unei diagrame de clase.



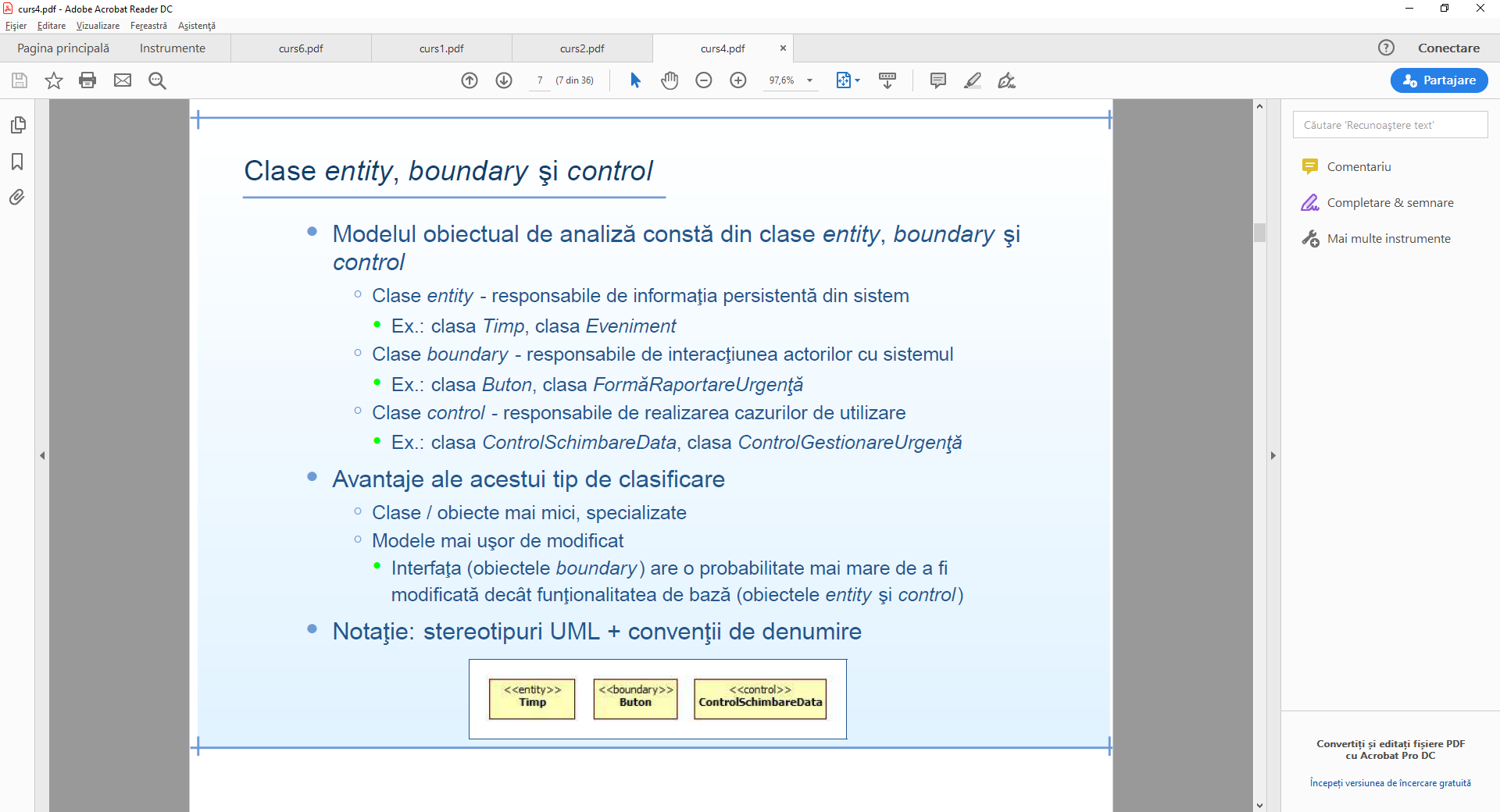
A4. Explicati modul in care numele de roluri si multiplicitatile specificate la nivelul diagramei de clase influenteaza codul sursa aferent acelei diagrame. (1.5p)



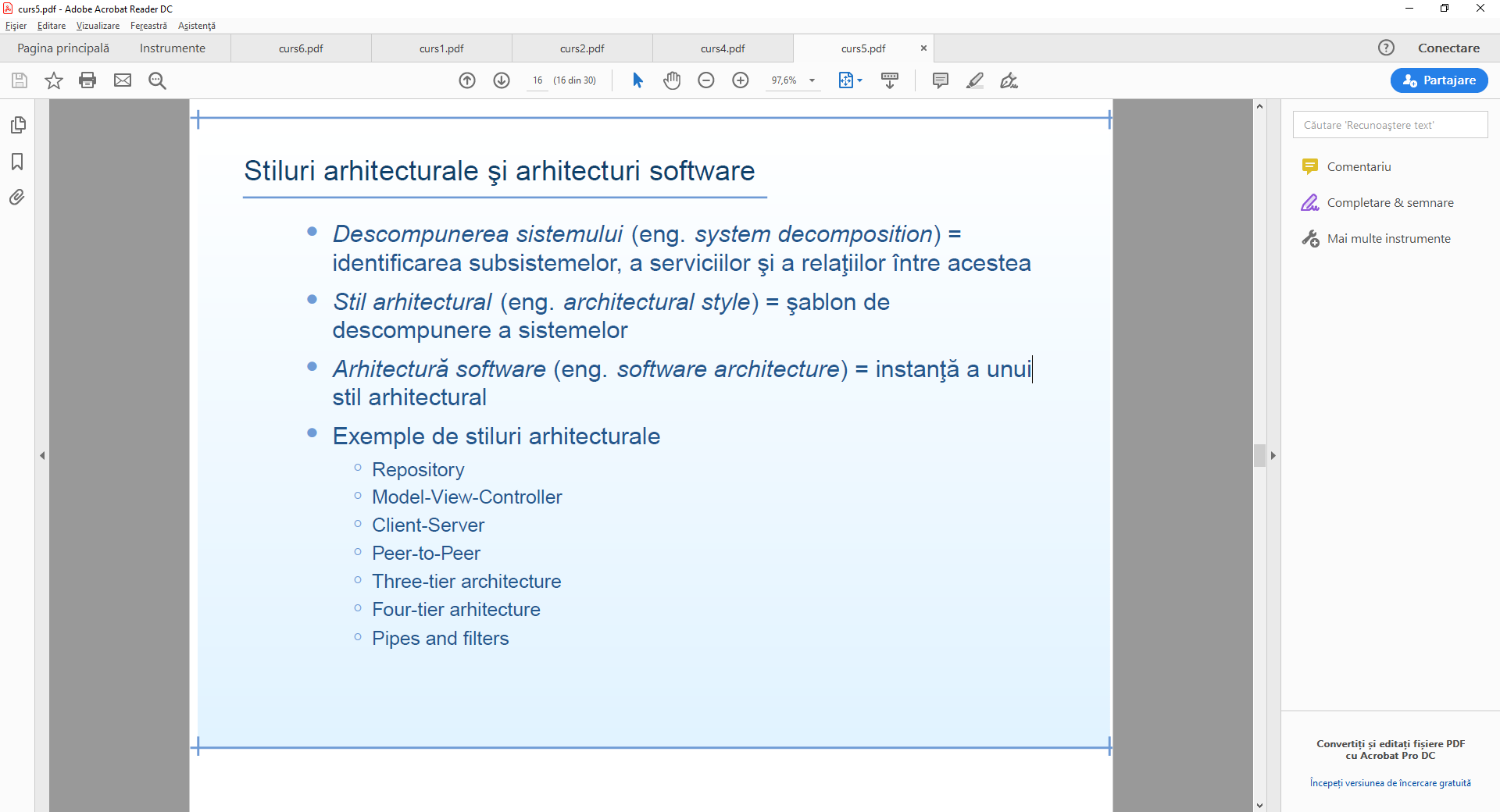
1..\* / 1 / \* va determina daca atributul din clasa este simplu sau colecti

Numele (+rapoarte / +initiator) din capetele sagetilor determina numele atributelor din clase.

A5. Explicati semnificatia stereotipurilor <entity>, <boundary> si <control>. (1.5p)



A6. Dati trei exemple de stiluri arhitecturale. (1.5p)



B (60%). Se considera urmatoarea problema. Se cer urmatoarele:

B1. Identificati actorii si cazurile de utilizare aferente acestora. (1p+1p)

Actorii sunt:

- CLIENT

- OPERATOR

- SGBD

Cazurile de utilizare sunt:

- CUMPARARE PRODUS

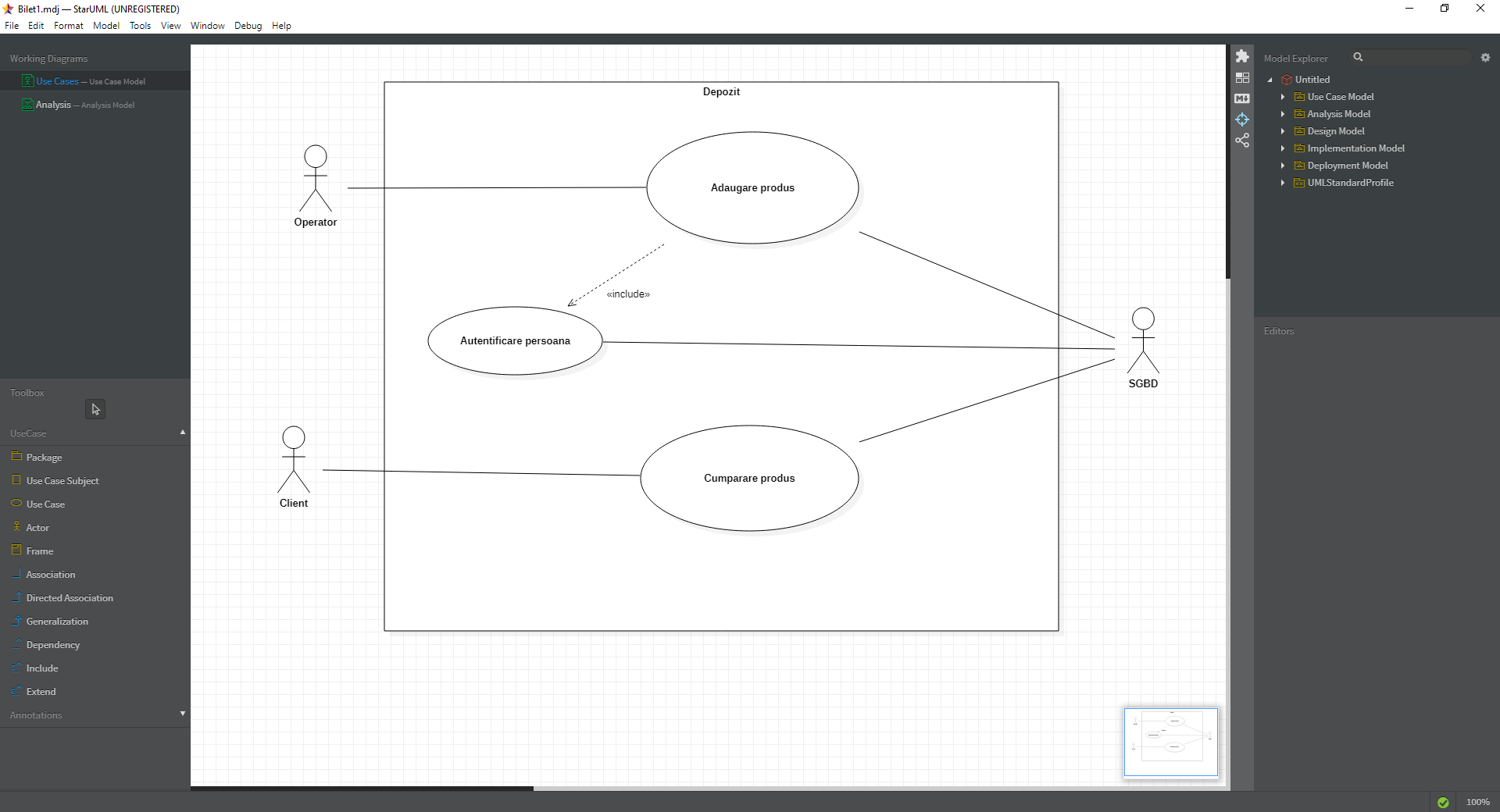
- ADAUGARE PRODUS

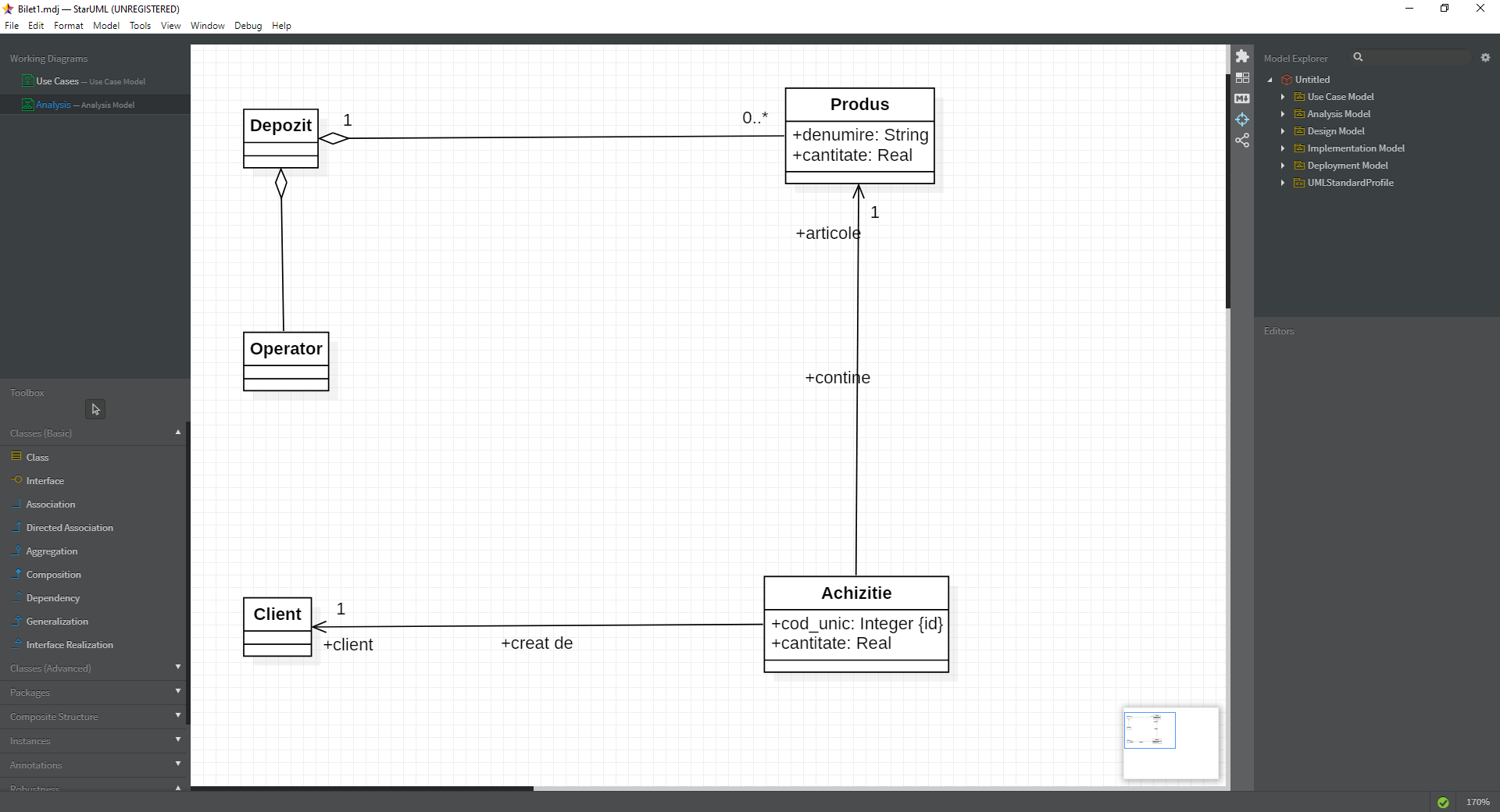
B2. Trasati diagrama cazurilor de utilizare. (2p)

B3. Identificati conceptele, atributele acestora si relatiile dintre ele. (1p+1p+1p)

B4. Trasati diagrama de clase, care sa includa elementele de model identificate la punctul B3. (2p)

Depozit de produse La un depozit de produse se foloseste o aplicatie pentru a tine evidenta produselor din depozit. Pentru fiecare produs se pastreaza denumirea si cantitatea disponibila. Cand un client vine sa cumpere produse, se alege produsul din lista celor existente, se introduce cantitatea dorita si se apasa butonul "CUMPARARE". Dupa apasare cantitatea disponibila se actualizeaza. Operatorul poate sa adauge noi produse. Pentru un nou produs se introduce denumirea si cantitatea disponibila. Dupa adaugarea unui nou produs lista produselor se actualizeaza. La pornirea aplicatiei informatiile despre produse se citesc dintr-o baza de date relationala, iar la inchiderea aplicatiei informatiile din baza de date sunt actualizate.

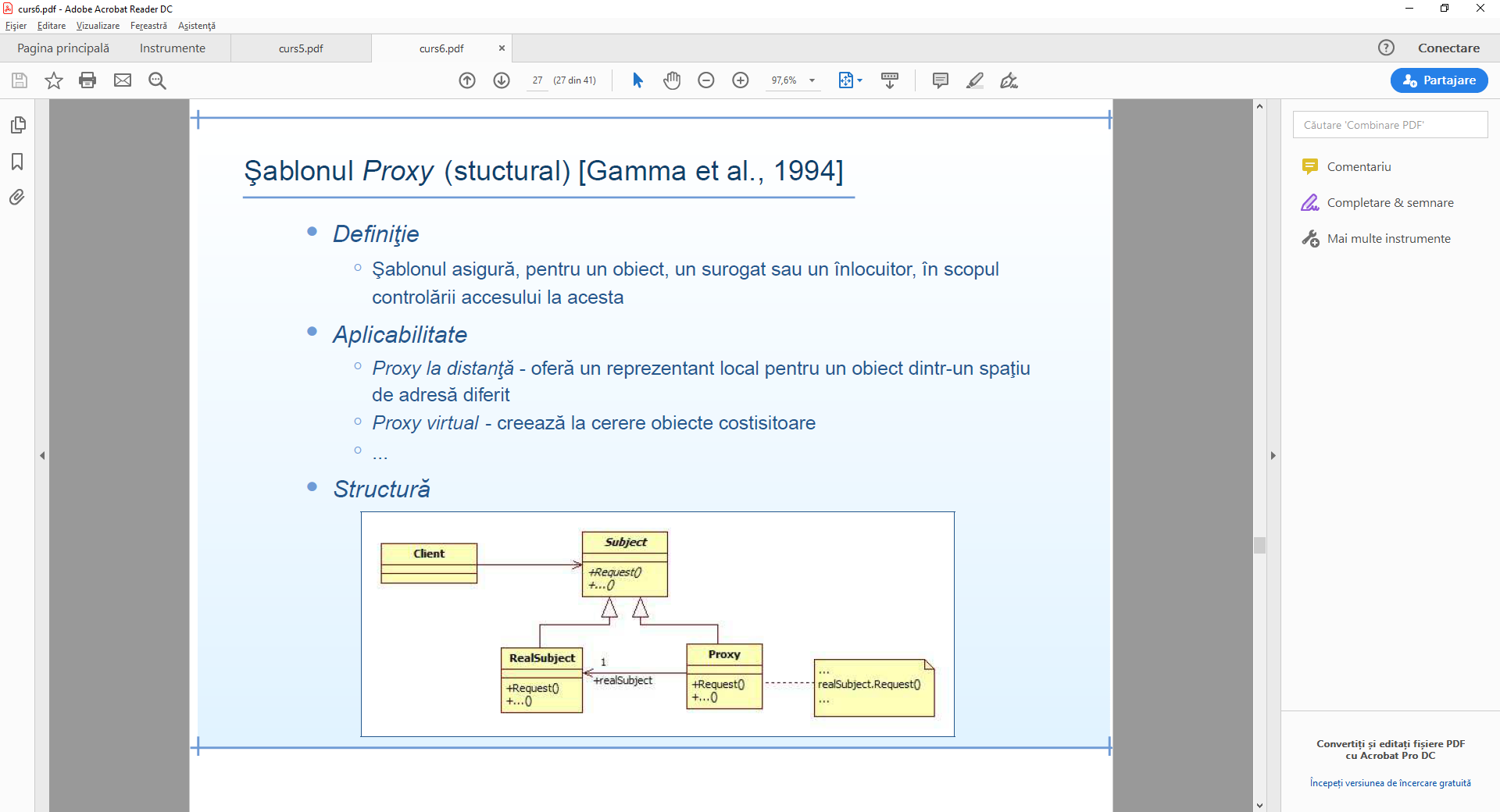




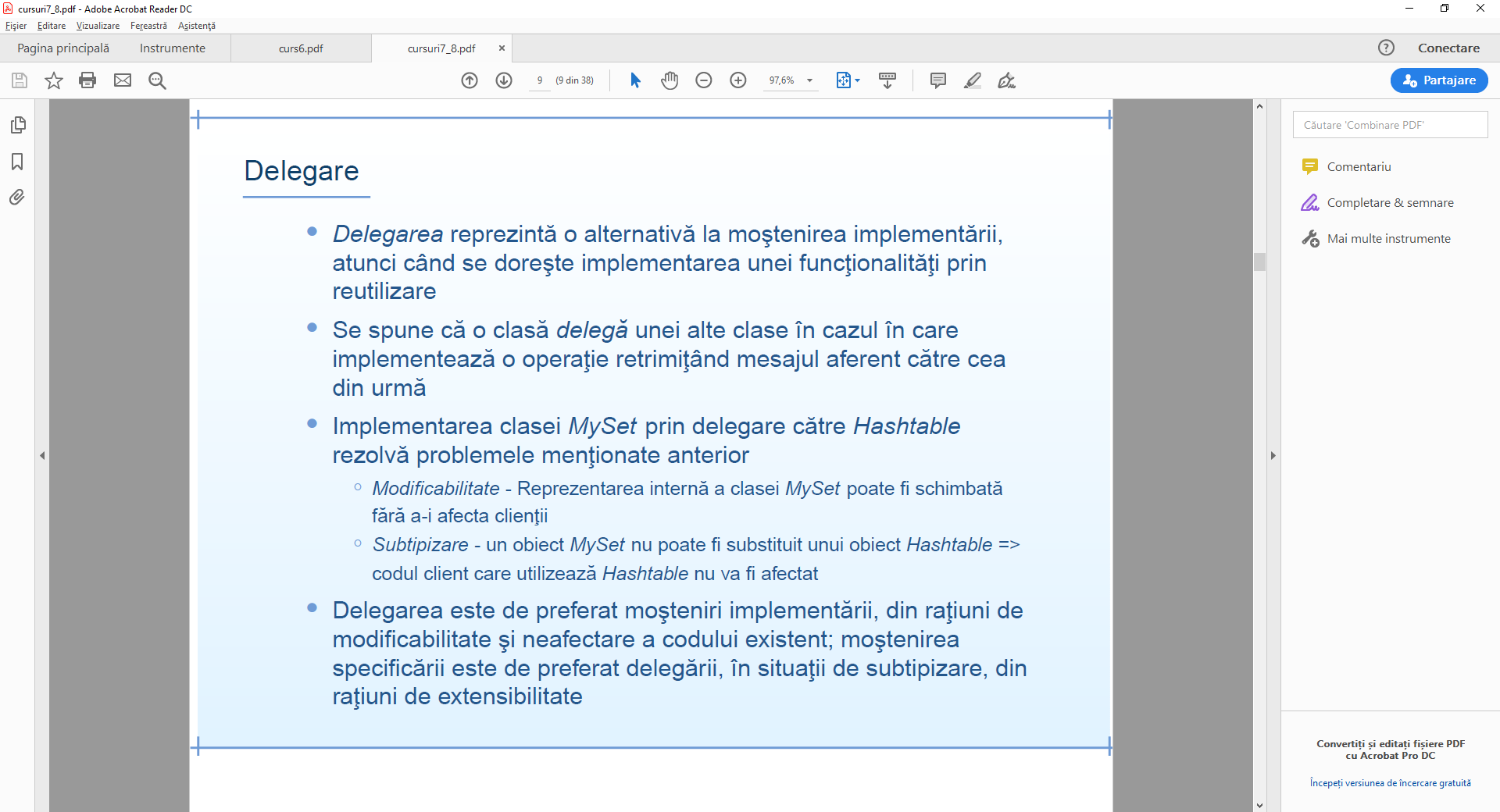
BILET 1 ( FINAL )

C (40%). Raspundeti succint la urmatoarele cerinte/intrebari

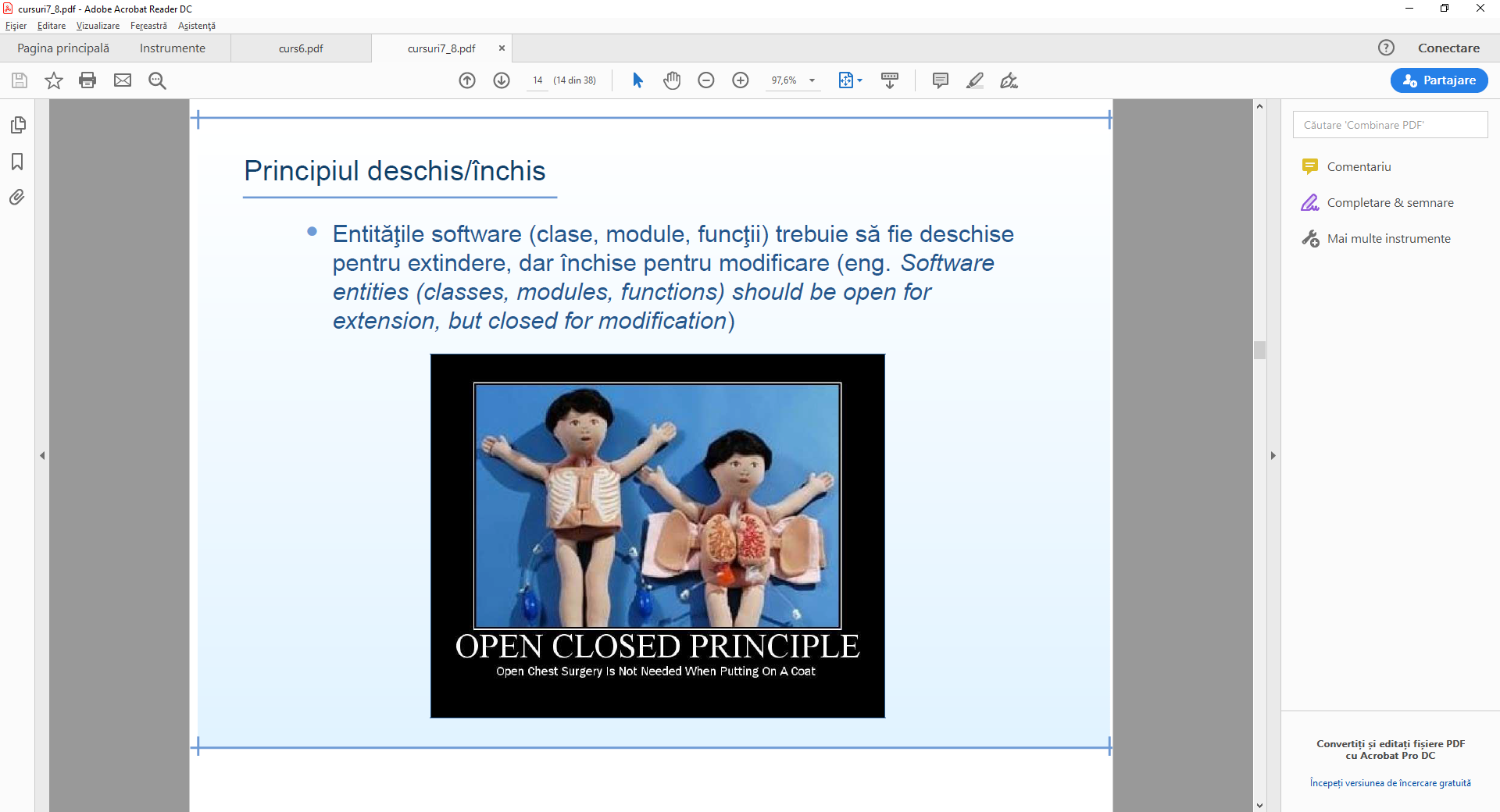
C1. Justificati utilitatea sablonului de proiectare Proxy. (1.5p)



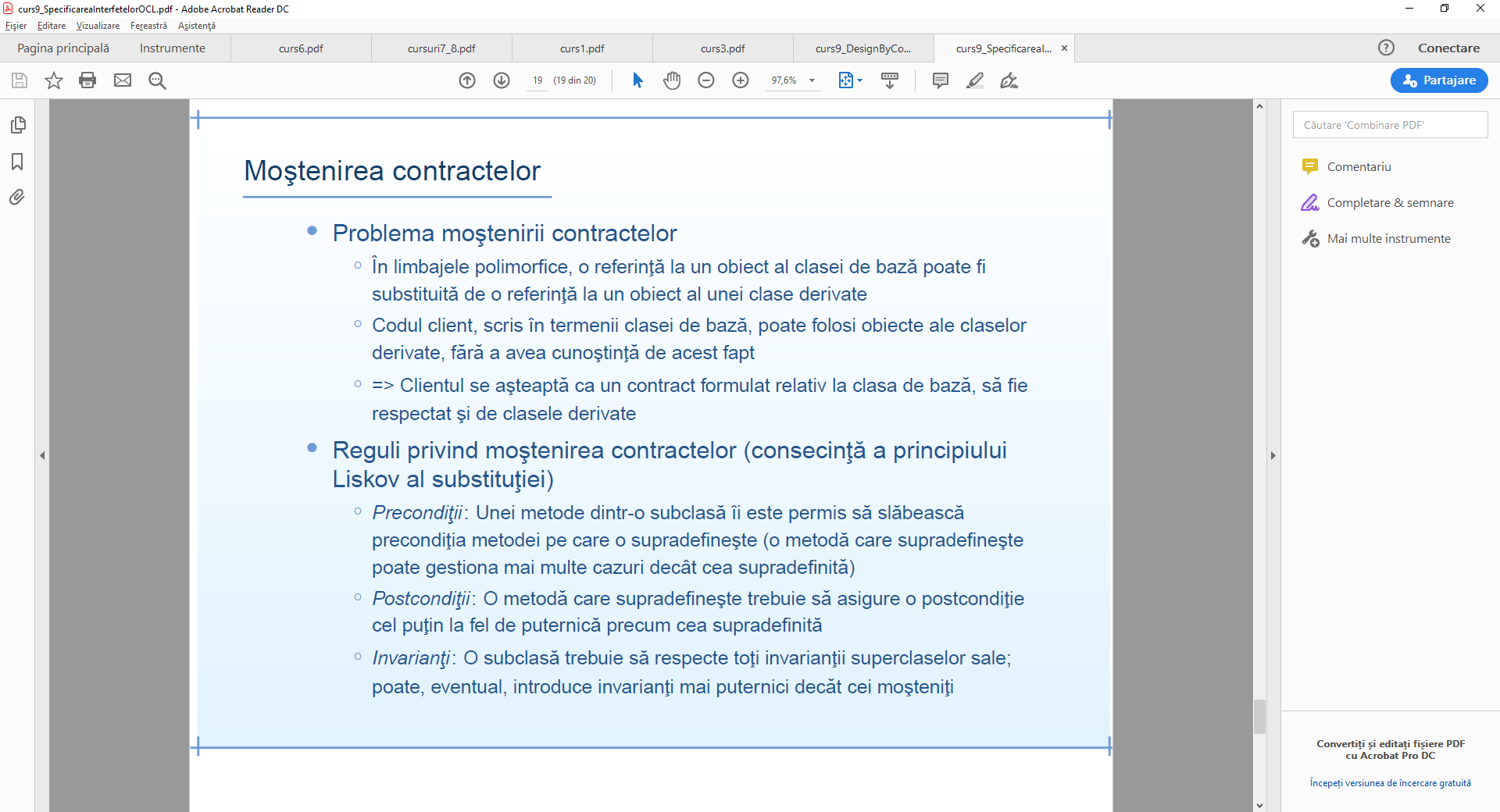
C2. Definiti conceptul de delegare. (1.5p)



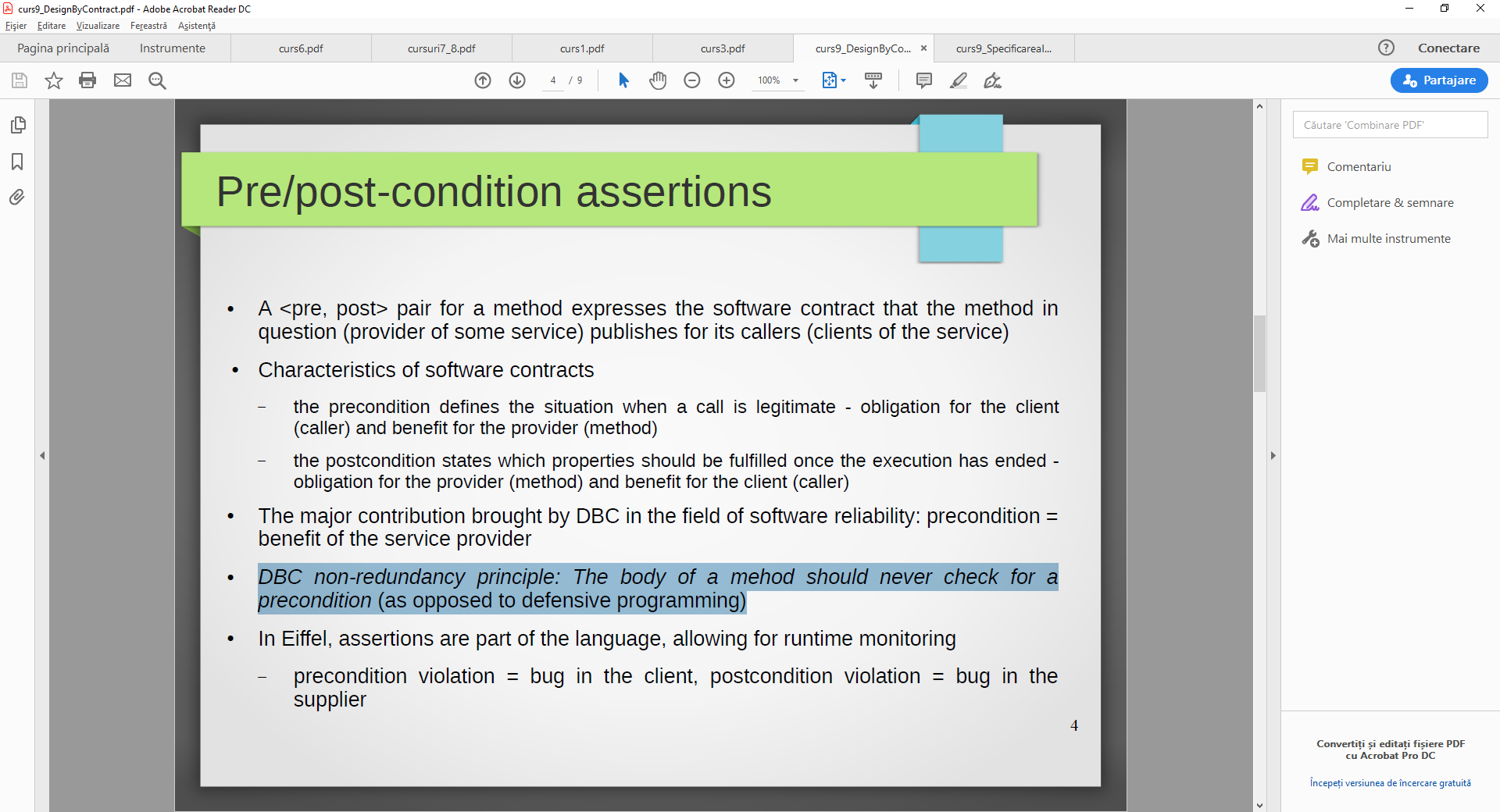
C3. Enuntati si explicati, pe scurt, principiul deschis/inchis (open closed principle). (1.5p)



C4. Enuntati regula privind mostenirea contractelor referitoare la preconditii. (1.5p)



C5. In ce consta principiul nonredundantei in contextul Design by Contract? (1.5p)

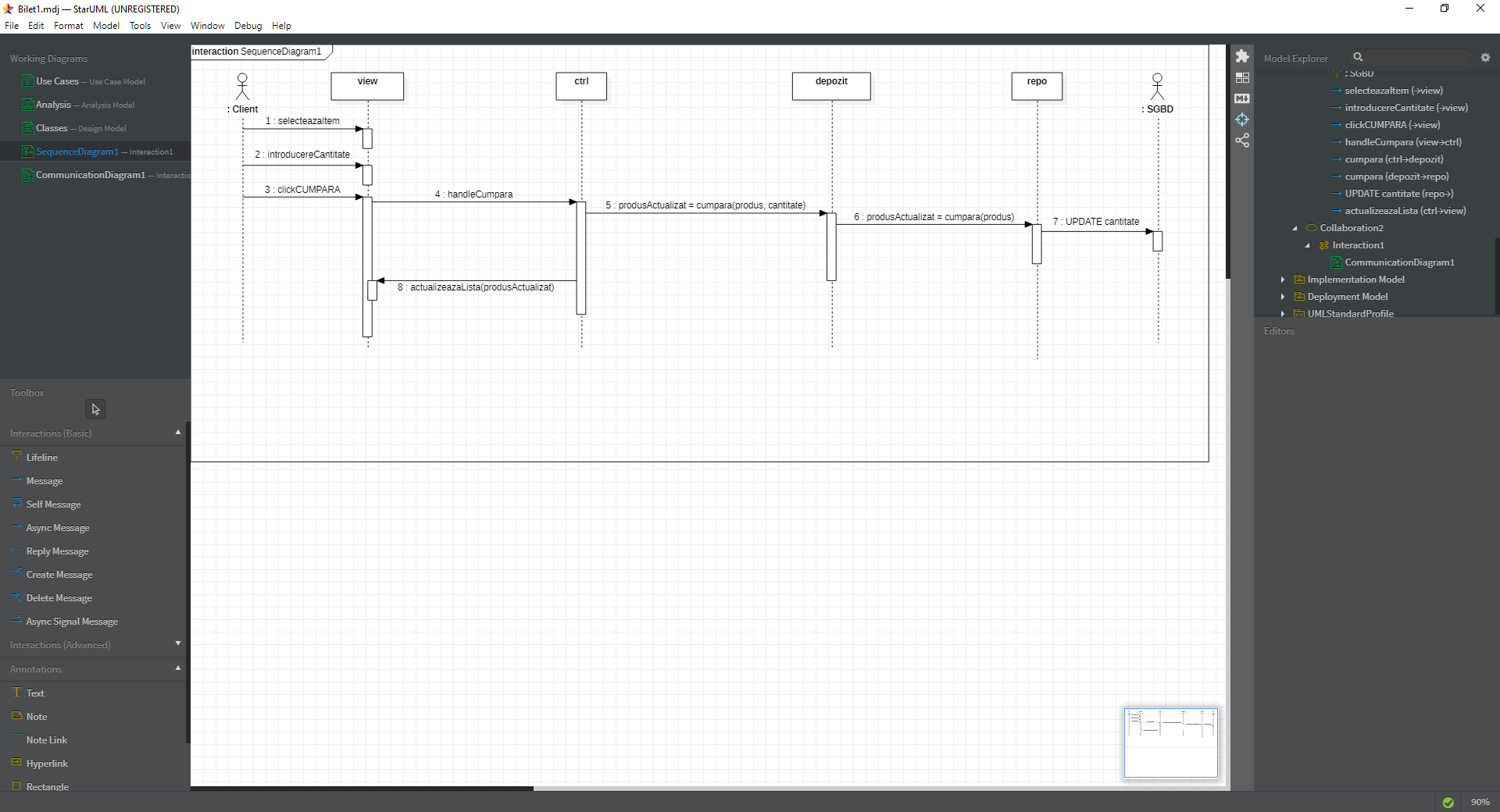


C6. Folosind UML si OCL, dati un exemplu de invariant in contextul unei clase, care sa foloseasca unul dintre iteratorii pe colectii. (1.5p)

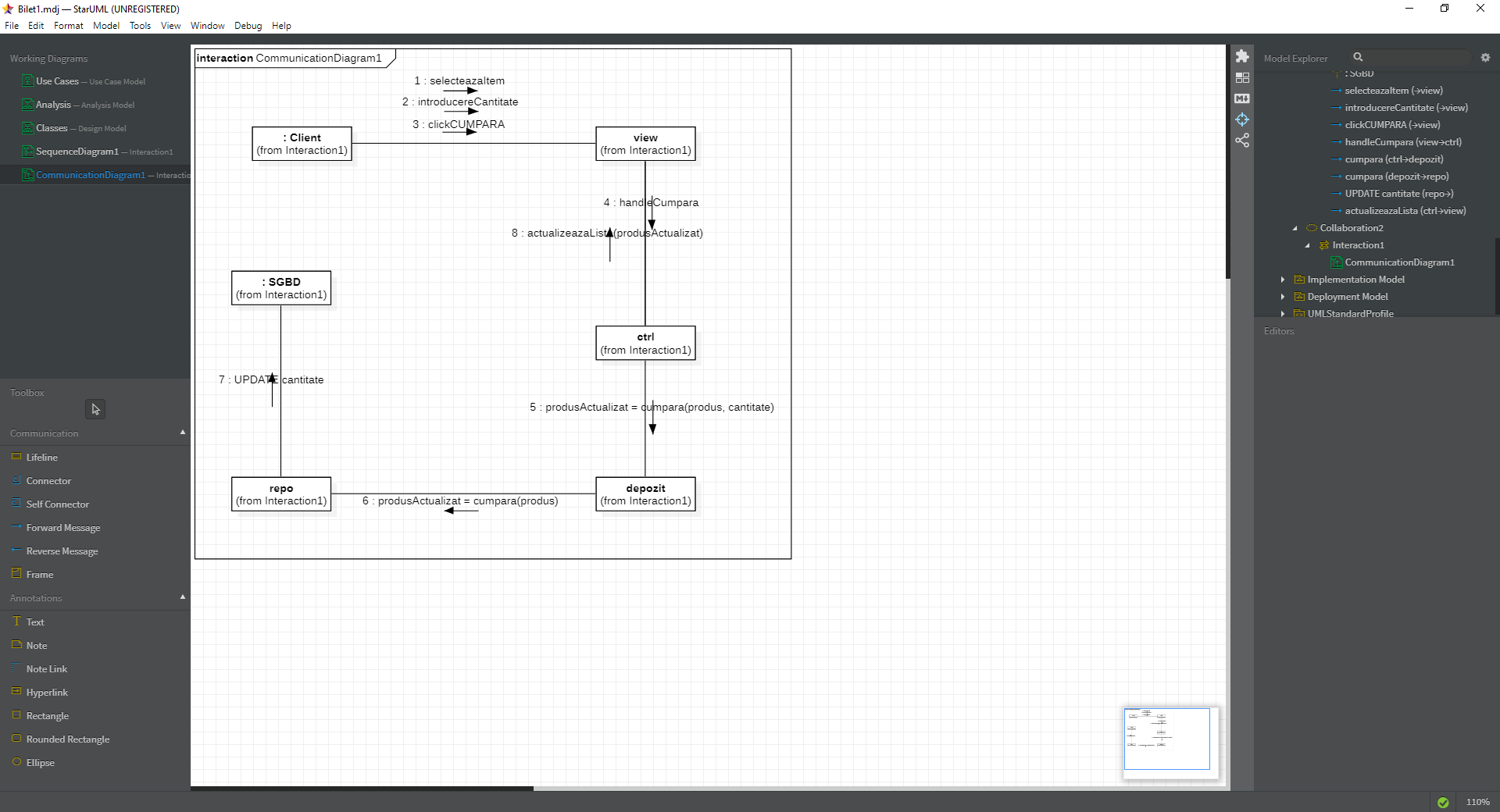
(pass)

D (60%). Se considera problema de la punctul B (verso). Se cer urmatoarele:

D1. Trasati diagrama de secventa pentru scenariul ideal al procedurii de CUMPARARE. (2p)



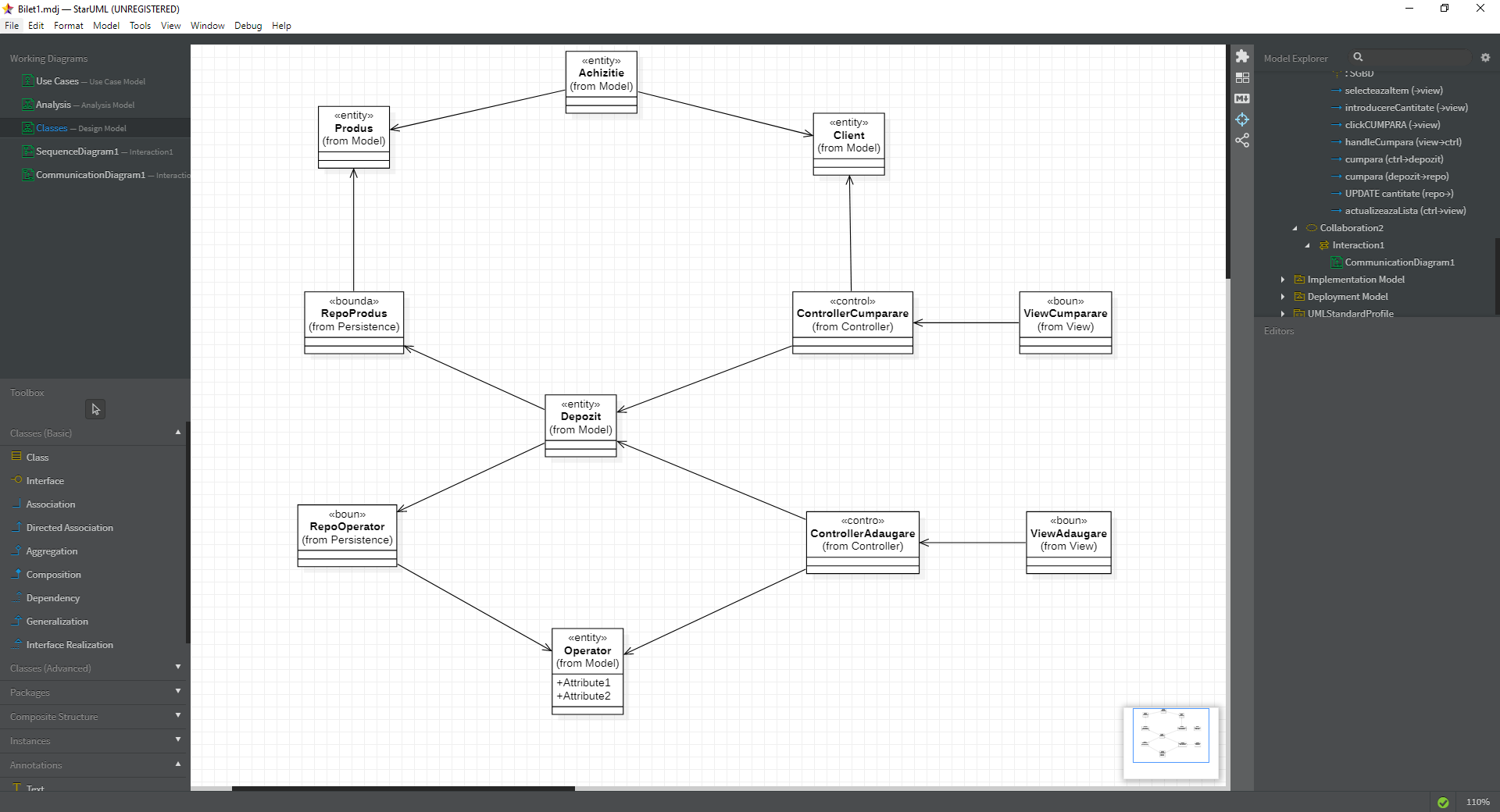
D2. Trasati diagrama de colaborare (comunicare) pentru acelasi scenariu. (2p)



!!! La evaluare 1.5p sunt alocate pentru concordanta dintre cele doua diagrame de interactiune.

(DISCLAIMER: NU ESTE COMPLET (posibil sa nu fie 100% corect din cauza incompletitudinii))

D3. Trasati diagrama detaliata de clase, care sa includa şi noile clase identificate la punctele D1 si D2. Aceasta diagrama va trebui sa includa atat atributele si metodele claselor cat si relatiile dintre aceste clase. (2p)



!!! La evaluare 1.5p sunt alocate pentru concordanta dintre diagrama de la D3 si diagramele de interactiune. (1p din oficiu)