

## Berarie

Sa se creeze o aplicatie pentru gestiunea produselor dintr-o berarie.

Se va utiliza clasa *Bere* (*nume: string, tip: string, pret: float*). Perechea (*nume, tip*) este unica.

Clasa *Bere* va expune o metoda

*get\_id(): string*

care va returna un string format din *nume* si *tip* despartite printr-un spatiu.

Informatiile sunt stocate in fisierul "produse", astfel:

<*nume*>, <*tip*>, <*pret*>

Heineken, blonda, 7

Ciuc, blonda, 5.5

Ciuc, nefiltrata, 5

Lefe, bruna, 9

Lefe, blonda, 10

Ciuc, bruna, 4

....

Se va folosi arhitectura stratificata. Se vor utiliza clasele: *Console, Controller, Repository*.

Clasa *Repository* va expune metodele

*get\_all(): Bere[]*

*update(bere\_id: string, bere: Bere)*

care va modifica in fisier attributele inregistrarii indentificate prin *bere\_id* cu attributele instantei *bere*.

La pornirea aplicatiei se vor afisa optiunile:

1 - Update pret

2 - Discount

x - Exit

Optiunea 1 - Update pret: se citeste un nume de bere si un intreg, x, reprezentand un procent. Sa se majoreze preturile tuturor berilor avand numele dat cu x%. Imediat dupa executia acestei operatii modificarile se vor vedea in fisier.

Optiunea 2 - Discount: se citeste un tip de bere; preturile fiecarei beri avand tipul citit va fi deveni egal cu minimul dintre preturile berilor cu acel nume. Imediat dupa executia acestei operatii modificarile se vor vedea in fisier.

Exemplu: daca se citeste tipul "blonda" atunci berea Ciuc, blonda va avea pretul 4 si berea Lefe, blonda va avea pretul 9.

Barem:

Oficiu: 1p

Specificatii si teste: 2p

Arhitectura: 1p

Stil: 1p

Optiunea 1: 2p

Optiunea 2: 3p

