

Proiect POO

Administrare bar

Cațaron Andrei
grupa 211

Părțile 1 și 2

În acest punct proiectul conține implementările a 6 clase: 3 clase de baza: **Item** (clasă abstractă), **Ingredient** și **Comanda**. Clasa **Item** este moștenită de clasele derivate **Sticla_Vin** și **Cocktail**. Clasa **Sticla_Vin** este la rândul ei moștenită de **Pahar_Vin**.

Produsele barului vor fi obiecte derivate din **Item**. Pe lângă variabile, constructori, setteri și getteri clasa abstractă conține două funcții pure virtuale

```
virtual ostream& print(ostream&) = 0;\\  
virtual bool contine_alcool() = 0;
```

Prin intermediul primeia se supraîncarcă operatorul << doar în clasa abstractă (funcția print se definește în fiecare clasă derivată).

În clasa **Sticla_Vin** pe lângă variabile, constructori, operator de atribuire și definițiile funcțiilor virtuale sunt supraîncărcarea operatorii + = și ++ care modifică variabila *vechime*. Clasa **Pahar_Vin** conține un singur constructor care primește ca argument un obiect de tipul **Sticla_Vin** (ale cărui caracteristici le preia) și o metodă care calculează prețul.

Clasa **Ingredient** conține trei variabile dintre care una de tip **char***. Pe lângă implementările obișnuite, este supraîncărcarea operatorul < deoarece **Ingredient** va fi folosit de post de cheie într-un **map**.

Clasa **Cocktail** dorește să implementeze faptul că cocktail-urile se obțin prin amestecarea unor băuturi mai simple de tip **Ingredient**. Astfel, singura variabilă este un **map<Ingredient, int>** care va conține ingredientele

folosite cu valoarea procentul utilizat în amestec. **Cocktail** mai are implementate câteva metode care calculează diverse valori pornind de la **map**-ul folosit și supraîncărcarea operatorului `[]` pentru a extrage elemente din **map** (poate fi folosit de exemplu pentru a afișa ingredientele din **map**).

Clasa **Comandă** conține un **vector<Item*>** în care se rețin produsele comandate de client. Pentru a adăuga cu ușurință un nou produs pe comandă a fost supraîncărcat operatorul `+=`. De asemenea au mai fost supraîncărcați și alți operatori unari și binari.

Partea 3

Pe lângă mici modificări aduse claselor deja implementate, a fost adăugat în proiect un meniu în consolă destinat interacțiunii cu utilizatorul. Pentru gestionarea barului, au fost implementate un **list<Item*>** **meniu**, un **set<Ingredient*>** **ingrediente** și un **vector<Comanda*>** **comenzi**. Atât cu meniul cât și cu lista de ingrediente se poate interacționa din consolă sau din fișiere externe.

În implemetarea meniului au fost folosite elemente noi de programare precum: tratarea excepțiilor, scrierea și citirea din fișier, funcția **find_if** din **algorithm** împreună cu expresii lambda, **dynamic_cast**, diverse operații cu pointeri.