

# Домашна работа №:2 по Обектно-ориентирано програмиране

Автор: Ангел Димитриев

## 1 DrinkBook

В София е пълно със заведения, които всеки може да посети, но те били различни видове. Столичани много обичали да ги обикалят, но много се ядосвали когато попаднат в заведение, което не отговаря на вкуса им. След многото успешни ремонти в центъра на града(най-вече на графа), столична община е решила да насочи вниманието си и към купонджиите и да им осигури система, с която да могат бързо да мигрират м/у заведенията. За ваш късмет, вие сте избрани за проекта.

Ще имате **потребител**, който има име, години, бюджет, брой уискита, които иска да изпие, брой водки, които иска да изпие, и предпочитана музика. Потребителите ви имат 4 вида предпочитания за музика : **Поп-фолк, Хаус, Рок и "Слушам всичко"**.

Ще има 3 вида заведения:

**Поп-фолк заведение**, което има име, капацитет от 70 души, цена на водката  $\geq 20$  лв, цена на уискито  $\geq 35$  лв и изпълнител за вечерта.

**Хаус заведение**, което има име, няма капацитет за хората(могат да наблъскат и 1000 души в една зала), цена на водката  $\geq 30$ лв, цена на уиски  $\geq 40$ лв и брой DJ за вечерта(очевидно цяло число).

**Рок зведение**, което има име, капацитет от 30 души, цена на водката  $\geq 40$  лв, цена на уискито  $\geq 30$  лв.

Всеки обичал да посещава заведение, което е с неговата музика, но когато попадне в заведение, в което пускат друго, можело да стане лошо(съответно ние не трябва да го допускате).

**Феновете на рок** можели да се прежалят да отидат на хаус, но се ужасявали от мисълта да отидат на чалга. Смятали посетителите на такъв вид заведения за пропаднали, и не желали да се смесват с "простолюдието".

**Хаус-аджиките** пък харесвали да ходят на поп-фолк заведения, понеже лесно се запознавали с момичета там, но не уважавали рок заведенията, понеже редовно си тръгвали набити от там(а и не им харесвали черепите по стените).

За **поп-фолк феновете** не представлявало проблем да се забавляват на рок музика, но хаус музиката и хаус заведенията не били по вкуса им и постоянно измисляли подигравателни шеги за момичетата по тези заведения.

За **"Слушам всичко"** , естествено, музиката не представлявала проблем и се забавлявали навсякъде.

Трябва да направите колекция от заведения, в която ще можете да добавяте/премахвате заведение. Може да имате произволен брой заведения. Ще трябва да можете и да добавяте потребител в заведение и функцията

трябва да връща **ИСТИНА**, ако потребителят е влезнал и **ЛЪЖА**, ако е отказал. Ще трябва да можете и да премахвате потребител от заведение. Потребителят влиза в дискотеката, когато:

- Харесва музиката, която пускат в дискотектата.
- Има място в дискотеката(не е препълнена, понеже няма да го пуснат)
- Цените му позволяват да изпие толкова, колкото иска.
- Има навършени 18 години. Ако няма навършени 18 години, могат да го пуснат единствено в поп-фолк заведение, но тогава бюджетът му спада с 20лв.(трябва да ги даде на охраната, понеже си е "забравил"личната карта).

---

```
{
  //Example
  Clubs d;
  d.addFolkClub("33",10,40,"Galena");//False (the Vodka is too cheap)
  d.addFolkClub("33",40,40,"Galena");// Much better
  d.addHouseClub("Yalta",50,50,2); // 50 for whiskey,50 for vodka and
    2 DJs this night.

  User u("Ivan",18,180,2,2,"House");
  //Name: Ivan, Age: 18, Cash: 180, Vodkas:2, Whiskeys: 2, Music: House

  d.addToClub(u,"Yalta");// not enough cash for 2 vodkas and 2 whiskeys
  d.addToClub(u,"33"); // True
  d.removeFromClub("Poli","33");//Flase. No Poli in 33 tonight
  d.removeFromClub("Ivan","33");//True
}
```

---

## 2 Логически изрази

Реализирай клас "**BooleanExpression**",който представлява клас за работа с **произволни изрази** от съждителната логика. Операциите, които трябва да поддържате, са  $\neg, \wedge, \vee, \Rightarrow, \Longleftrightarrow, \oplus$

Класът ви трябва да има:

- Функция, която принтира израза в четим формат.
- Функция, която връща колко променливи участат в израза.
- Функция **isTautology()**, която връща истина, ако изразът е тавтология(винаги е истина и не е възможно да бъде лъжа).
- Функция **isContradiction()**, която връща истина, ако изразът е отрицание (винаги е лъжа и не възможно да бъде истина).

- Функция **isContingency()**, която връща истина, ако изразът е условност. (не е тавтология и не е отрицание)

Не се изисква реализирането на потребителски вход (понеже `parse`-ването на такъв израз от текст е трудна работа). Но класът ви трябва да може да работи с всякакъв израз.

Нака обектът `expr` е  $p \wedge \neg p$ .

---

```
{  
    expr.isTautology();//false  
  
    expr.isContradiction();//true  
  
    expr.countVariables(); //1  
  
    expr.print(); //Example printing: a ^ (!a)  
}
```

---

**ВАЖНО.** Реализирайте задачите спазвайки добрите ООП практики. Решения, в които не са спазени тези практики, ще бъдат оценявани с 0 точки.