

Подготовка за контролно 1

ИС 2018г

Задача1.

Да се напише клас Account, който реализира потребителски акаунт, характеризиращ се със следните член-данни:

- `char* name` - име на акаунта (nickname);
- `char* password` - парола на акаунта (трябва да съдържа както букви, така и цифри, но САМО това и нищо друго, т.е. парола от вида *aaa?#b2* не е коректна, а парола от вида *aaabcd2cc3f* - е коректна);
- `int lvlOfPrivilege` - ниво на привилегии, които има акаунта, те могат да бъдат
 - 1 - за обикновен акаунт с привилегии като гост (Guest);
 - 2 - за стандартен съществуващ акаунт (Member);
 - 3 - за акаунт с привилегии на администратор на сайта (Admin);

Да се реализират:

- Канонично представяне;
- Функция `bool correctPassword()` - проверява дали паролата отговаря на посоченото условие.
- Функция `bool validate(char* name, char* password)` - която проверява дали подадените параметри представляват данните на конкретния акаунт;
- Функция `int getPrivilege()` - която връща нивото на привилегия на конкретния акаунт;
- Функция `void createAccount(char* name, char* password, int lvl)` - която дефинира данните на акаунта;
- Бонус: `void serialiseInTextFile(char* fileName)`, който записва акаунта в текстов файл с име, подадено като аргумент;

Задача2.

Да се напише клас Site, който реализира потребителски сайт, за който потребителите могат да създават собствен акаунт, а предназначението му е да съхранява проекти, създадени от потребителя. Класът се характеризира със следните член данни:

- Project* projects - масив с променлива дължина, пазещ добавените проекти;
- Account* accounts - масив с променлива дължина, пазещ добавените акаунти.

Забележка: Веднъж създадени с дадена големина, масивите нямат възможността да се преобразмеряват т.е. не е необходима функция `resize`;

- `int access` - текущо ниво на достъп, зависещо от това, дали в сайта е логнат потребител със свой акаунт. Първоначално никой не е логнат и достъпът е в режим `Guest`. (Виж стойностите за достъпа от Задача 1)

Да се реализират:

- Канонично представяне;
- Функция `void addAccount()` - която създава и добавя акаунт като комуникира с потребителя (подканва го да въведе необходимите данни);
- Функция `void deleteAccount(char* nick)`;
- Функция `void addProject()` - която създава и добавя проект само ако потребителят има право да го направи - потребителят трябва да бъде `Member` или `Admin`;
- Функция `void deleteProject(int ID)`;
- Функция `void printProjects()` - която отпечатва на екрана всички видими от текущия потребител проекти (сравнява привилегиите);
- Функция `void LogIn()` - опция за логване в сайта, която подканва потребителя да въведе име и парола. При съвпадение с някой от вече създадените акаунти, да се промени нивото на достъп на сайта `access` в зависимост от привилегията на акаунта;
- Функция `void LogOff()` - опция за изход от акаунт, която връща член-данната `access` към подразбиращата ѝ се стойност ;
- Функция `sortProjects()` - сортира проектите по ID;
- Оператори:
 - `+` - добавя проект, като резултатът е нов сайт;
 - `+=` - добавя проект към текущия сайт;
 - `-` - премахва акаунт, като резултатът е нов сайт;
 - `-=` - премахва акаунт от конкретния сайт;
 - **За вход и изход**