

Домашна Работа

Лекция 6

Абстракция. Интерфейси. Абстрактни методи и класове. Предефиниране (override) на методи и полиморфизъм.

- Моля прочетете отново лекцията преди да започнете да решавате домашното.
- I. Да се създаде интерфейс User –
 - юзърите трябва да могат да се логват(метод, който променя state-а на логнат)
 - да се логават(метод, който променя state-а на логнат)
 - да се взима username на всеки юзър(метод, който връща String съдържащ username-а, на потребителя)
 - да се взима датата, на която се е регистрирал(метод, който връща String съдържащ датата, на която потребителя е бил създаден)
- II. Да се създаде интерфейс AdminUser, като той е и User – админ юзърите ще могат да правят всичко, което и нормалните, но също така, ще могат и да –
 - Изтриват юзър, който ще намират на базата на неговия username
 - Ще могат да създават нови юзъри и от тях ще зависи дали те ще могат да са админи или не(метод, който приема два параметъра, за username и дали е админ) и връща, като стойност създадения нов User.
 - Ще могат да преглеждат всички потребители, метод който принтира цялата информация за всеки един потребител в конзолата(без значение дали е админ или не)
- III. Да се създаде abstract-ен клас AbstractUser, който е User (помислете какво бихте изнесли в него, какво общо има между потребителите и т.н.)

Жокер: Относно това как да пазите, registrationDate-а на всеки потребител използвайте класът java.util.Date в Java (<https://www.mkyong.com/java/java-how-to-get-current-date-time-date-and-calender/>)
- IV. Да се създаде клас DataBase, който съдържа в себе си масив от User[]-и
 - Има конструктор, който приема колко голям да е масива
 - Има гетър, който ви дава възможност да вземете всички потребители
- V. Да се създаде UserImpl клас, който е User(AbstractUser).
- VI. Да се създаде AdminUserImpl клас, който е User(AbstractUser) и AdminUser

- Всеки AdminUser ще си има собствена база от данни(DataBase) с хардкодната стойност по подразбиране за големината на броя потребители, които могат да бъдат създавани.(например всеки AdminUser има възможност да създава по максимум 10 потребителя).
- Методът, който ще създава потребител, ще се съобразява с това дали да създаде нормален потребител или админ, в зависимост от подадените параметри.
 - Ако вече има съществуващ потребител с такова име, той НЕ се добавя и се изписва съобщение в конзолата, че е вече съществуващ
 - Ако базата данни е вече пълна, ще изписва съобщение, че няма повече място за нови потребители
- Методът, който ще изтрива потребители от базата, просто ще го прави null.
 - Ако базата е празна и се опита да изтрие потребител, ще изкарва съобщение в конзолата, че няма какво да се трие от празна база данни.

VII. Създайте Demo клас с main метод, в който:

- Създайте един AdminUser
- Създайте N на брой потребителя извиквайки метода за създаване на потребител на AdminUser(и админи и нормални потребители)
- Поиграйте си и с граничните стойности дали са покрити(т.е. да не може да създадете повече потребители от колкото са заложили, и дали можете да изтриете потребител от празна база данни)