Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

Домашна работа 3

курс Обектно-ориентирано програмиране
за специалност Информатика
Летен семестър 2017/18 г.

FMIBOOK

От скоро разчулите се новини стана ясно, че често се случва социалните мрежи, които повечето от нас използват, да продават информацията от нашите профили на трети лица. Маркони иска да направи нова социална мрежа, в която всички данни са защитени от външния свят. Понеже не разбира особено много от програмиране, Маркони, на кратко Марк, си търси кой да му помогне. Дочул, че във ФМИ имало добри студенти, и че им трябва трето домашно. Затова решил да им възложи част от проекта си, за да види как ще се справят и дали ще ги наеме за останалата част.

Марк желае да има **три вида потребители** с различни права - обикновени, модератори и администратор. Всеки потребител си има **уникален прякор** (nickname) и възраст в години.

- Обикновените потребители могат да **добавят** публикации и да **премахват своя** публикация. Могат и да си сменят прякора, стига новият вече да не е зает.

- Модераторите могат всичко, което и обикновените потребители, но също така могат и да премахнат, която и да е публикация, **без значение на кой потребител**, както и да **блокират/разблокират** даден потребител. Блокиран потребител може да чете постове, но не може да публикува. Един модератор може да блокира друг, но ако не е съвестен той може сам да се разблокира.

- Администраторът може да прави всичко това, но си има и своя работа - да се грижи за **добавянето и премахването на потребители** в социалната мрежа. Единствено той може да добави в мрежата потребител и да укаже дали е модератор или обикновен потребител. Освен това има правото да премахне

всеки един от останалите потребители. При премахването на потребител от социалната мрежа се изтриват всички негови публикации.

Понеже Марк настоява да има разнообразие, трябва да се поддържат **три вида публикации**. Всяка една от тях си има **съдържание** и **уникален номер**, с който може да се цитира. Освен това всяка публикация трябва да може да се визуализира в HTML файл.

- Публикация-картинка. При добавянето ѝ потребителят задава абсолютен път до дадено изображение. Когато друг потребител иска да разгледа поста, за него се създава HTML файл, който визуализира това изображение. Може да прочетете как става създаването на HTML с изображение на: https://www.w3schools.com/html/html_images.asp

- Публикация-линк. При добавянето му, потребителят задава някакъв URL и описание. Публикацията отново ще е HTML файл, който съдържа хипервръзка с текст описанието, сочещ към подадения URL. Как да направите HTML хипервръзка може да прочетете тук.

- Публикация-текст. Тук потребителят подава просто текст и при прегледа на тази публикация се създава HTML файл, който съдържа само текста.

В социалната мрежа участват произволен брой потребители, както и модератори, но има само един администратор, който се задава при създаването на мрежата.

Системата ще работи посредством заявки, в следния формат :
<actor> <action> <subject>

Могат да бъдат изпращани следните заявки :

- Добавяне на нов потребител или модератор.
  Пример : Admin add_moderator Kiril 23 - добавя модератора Кирил на 23г.
  Admin add_user Vasil 30 - добавя обикновения потребител Васил на 30г.

- Премахване на потребител или модератор.
  Пример : Admin remove_user Vasil

- Блокиране / разблокиране на потребител или модератор.
  Пример : Admin block Kiril
  Kiril unblock Kiril - хм, този модератор не е много съвестен :-/

- Добавяне на нова публикация.
  Пример : Kiril post [image] `/home/kiril/img/hello.png`

Admin post [url] http://www.example.com някакво описание
Kiril post [text] Hello, world!

- Премахване на публикация.
  Пример : Kiril remove_post 8

- Потребител разглежда пост на друг потребител.
  Пример : Kiril view_post 10 - за Кирил се генерира HTML файл със съдържанието на публикация номер 10.

- По подаден потребител да се създаде HTML файл с всички негови публикации.
  Пример : Ivan view_all_posts Kiril

  *При всяка от заявките съобразете дали потребителят изпълнител (actor) има нужните права и възможности да извърши действието (action).

  **Изключение от формата на заявките правят следните две :

- Извеждане на статистики - колко обикновени потребителя и колко модератора има в момента, кой е потребителят с най-много постове, колко и кои потребители са блокирани в системата, кой е най-младият и кой най-възрастният потребител на мрежата.
  Пример: info
- Изход от програмата - quit

Направете прост (но използваем) конзолен интерфейс, който да предоставя описаните възможности. Пример за работа със системата е:

```
$: Admin add_user Pesho 21
Pesho created.
$: Pesho post [url] http://www.example.com Някакъв пример.
Post 0 created.
$: Pesho post [image] /home/pesho/img/hello.png
Post 1 created.
$: Admin add_moderator Gencho 28
Gencho created.
$: Gencho post [text] Hello everybody!
Post 2 created.
$: Pesho post [text] Hi, Gencho!
Post 3 created.
$: Pesho post [text] See my new bike:
Post 4 created.
$: Pesho post [image] /home/pesho/img/dzvera.png
```

Post 5 created.
$: Pesho post [text] egati i typoto kolelo e!!!
Post 6 created.
$: Pesho remove_post 6.
Post 6 removed.
$: Pesho post [text] egati i qkoto kolelo e!!!
Post 7 created.
$: Gencho post [text] Please do not use shliokavitsa!
Post 8 created.
$: Pesho post [text] Be ti li 6a mi kaje6!?
Post 9 created.
$: Gencho block Pesho.
Pesho blocked.
$: Pesho post [text] Na kyw mi se prai6!?
Post not created - user blocked!
$: Pesho post [text] Ej!!!!?
Post not created - user blocked!
$: Gencho post [text] Pesho, please, calm down!
Post 10 created.
$: Pesho post [text] Ti li 6a mi kave6!!!!?
Post not created - user blocked!
$: Admin *remove_user* Pesho.
Pesho removed.
$: Admin rename Pesho.
User Admin is now known as Pesho.
$: info
There are 2 users:
Pesho - Administrator, 0 posts.
Gencho - Moderator, 2 posts.
There aren't any blocked users.
oldest Gencho 28
youngest Gencho 28
$: Gencho remove_post 9
No such post!
$: Gencho remove_post 10
Post 10 removed.
$: Admin add_user Mimi 19
No such user : Admin!
$: Pesho add_user Mimi 19
User Mimi created.
$: Mimi post [image] /home/mimi/img/profile_pic.jpg
Post 11 created.
$: Gecho view_post 11

```
HTML view for post 11 created.
$: Mimi view_all_posts Gencho
HTML view for all Gencho's posts created.
$: delete system
Unknown command!
$: quit
```

Забележки и препоръки:

- Целият ви код трябва да спазва добрите ООП практики.
- Съветваме ви да реализирате по един клас за всеки тип потребител, по един за всеки вид пост, един за системата като цяло и ако прецените още един или повече, които да ви помогнат да решите задачата. Помислете какви взаимоотношения е добре да имат помежду си. Съветваме ви също да коментирате идеите си за ООП дизайн с някого от екипа, за да получите насоки и препоръки.
- Направете командите така, че да ви е лесно да работите с тях. Например може да добавите параметър какъв тип пост се прави (думите [image|text|url] в примера).
  Не е задължително вашия интерфейс да следва дадения пример, но крайната функционалност би трябвало да отговаря на зададената в условието.
- При всякакви въпроси и неясности по условието е желателно да пишете във форума в moodle, за да може всеки един от колегите (и от екипа, и от студентите) да е наясно какво се дискутира. Ако се забавим с отговора, молим да помолете някой от екипа да обърне внимание на поста ви.
- Не оставяйте решаването на задачата за последния момент. Това домашно е по-обемно и ще ви отнеме време, но ако подходите добре не е сложно. Също така ще ви е полезно, за да усвоите основни правила за ООП-проектирането.