



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

Домашна работа 1

курс Структури от данни и програмиране
за специалност Информатика
зимен семестър 2017/18 г.

Трябва да се направи браузър. На вас ви се пада честта да направите модула за табове на брауъра.

Всеки таб има URL (символен низ) и дата и време на зареждане на страницата (цяло число). URL-ът е този на текущо заредената в таба страница. Когато дадена страница се зареди в таба, за нея автоматично се запомня unix timestamp с времето и датата на достъпване. За да вземете timestamp за текущото време, може да използвате например функцията `time()` на C/C++.

В даден момент, потребителят на брауъра може да работи само с един от всички табове, който ще наричаме текущ. От текущия таб можем да се прехвърляме върху някой от непосредствените му съседни – или този преди него или този след него. Разбира се, това става само ако има такива (например, първият таб няма друг преди него). За целта трябва да се поддържат операции за преминаване назад (back) или напред (forward) спрямо текущия таб.

Приемаме, че в брауъра винаги има отворен поне един таб. Ако потребителят затвори последния таб, отваря се нов, в който се зарежда адрес “about:blank”. Когато браузърът се стартира, още преди потребителят да може да въвежда команди, в него веднага се създава един таб, отново с адрес “about:blank”. И в двата случая, този таб става текущ.

Ако потребителят отвори нов таб, той трябва да се добави непосредствено след текущия и на свой ред да стане текущ.

Ако затворим даден таб, текущ става този, който е непосредствено след него, а ако няма такъв – този преди него. Ако това е бил последният таб, важи правилото описано по-нагоре в текста.

Вашата задача е да реализирате отварянето и затварянето на табове, посещаването на страници и прехвърлянето от един таб на друг.

За решаване на задачата трябва да се реализира програма с команден интерфейс, в която могат да се въвеждат следните команди:

GO <url>	Командата зарежда страницата с URL <url> в текущия таб. Автоматично трябва да се обнови timestamp-а на таба.
INSERT <url>	Добавя нов таб след текущия, в който се зарежда страницата с URL <url>. За този нов таб автоматично се обновява неговия timestamp. Новият таб става текущ.
BACK	Потребителят се прехвърля на таба, който е непосредствено преди текущия. Ако такъв няма (в момента сме върху първия таб), не се случва нищо.
FORWARD	Браузърът отива на таба, който е непосредствено след текущия. Ако такъв няма (в момента сме върху последния таб), не се случва нищо.
REMOVE	Премахва текущия таб. Текущ става този след него. Ако такъв няма, текущ става този пред него. Ако това е бил последният таб, отваря се нов и в него се зарежда адрес "about:blank".
PRINT	Извежда на екрана всички табове. Форматът е следния: <URL> <timestamp> (Между двата елемента има един интервал) Редът, на който се намира текущия таб, трябва да започва със знак за по-голямо (>), например: > www.example.com 123456789 Страниците да се извеждат точно в реда, в който се пазят в брауъра.

Освен тях, може да се реализира и следната опционална команда (тя носи допълнителни точки към решението):

SORT <by>	Сортира всички табове лексикографски и прави текущ първия в наредбата. <by> може да приема две стойности: <ul style="list-style-type: none"> • URL – сортира табовете по URL, като ако има два с еднакъв URL, те се подреждат по timestamp; • TIME – сортира табовете по техния timestamp, като ако има два с еднакъв timestamp, те се подреждат по URL.
-----------	--

Когато реализирате програмата, изберете подходящ контейнер, от изучаваните в рамките на курса, в който да запазвате информацията за табовете и върху който да изпълнявате операциите. Тъй като първото домашно е върху материала за базовите структури от данни, достатъчно е да изберете измежду следните:

- масив
- динамичен масив
- свързан списък
- двусвързан списък
- цикличен списък
- стек
- опашка.

Ако реализирате и опционалната команда за сортиране, изберете подходящ алгоритъм за сортиране.

Изборът на структура от данни и начинът, по който ще работите с нея (вкл. сортирането, ако решите да го реализирате), трябва да бъдат съобразени с операциите, които ще извършва програмата ви. Помислете както за времевата, така и за пространствената сложност.

В задачата не може да се използват готови контейнери (като например тези от STL). Вместо това трябва да реализирате алгоритмите и структурите от данни сами. Не е необходимо да реализирате клас за контейнер (например клас за списък, масив, стек и т.н.), освен ако не ви трябва такъв. Достатъчно е да реализирате тази функционалност, която е нужна за решението, като обаче работите коректно с паметта и спазвате принципите за добър дизайн.

По-долу е даден пример за работа на програмата. Редовете започващи с \$ са за въвеждане на команди:

```

$ GO www.youtube.com/watch?v=dQw4w9WgXcQ
$ INSERT
www.google.bg/search?q=google+plz+help+me+solve+my+data+structures+task
$ INSERT
www.9gag.com/gag/aAV83g9/when-you-realise-today-is-the-homework-assignment-deadline
$ INSERT www.en.wikipedia.org/wiki/Data_structure
$ BACK
$ BACK
$ REMOVE
$ PRINT
www.youtube.com/watch?v=dQw4w9WgXcQ 150676023
>www.9gag.com/gag/aAV83g9/when-you-realise-today-is-the-homework-assignment-deadline 150670593
www.en.wikipedia.org/wiki/Data_structure 150700011
$ REMOVE
$ GO www.wikihow.com/Deal-With-Tons-of-Homework
$ FORWARD
$ GO www.susi.uni-sofia.bg
$ PRINT
www.youtube.com/watch?v=dQw4w9WgXcQ 150676023
>www.susi.uni-sofia.bg 163750320

```

В случай, че реализирате опционалната команда, тя може да работи например така:

```

$ GO www.youtube.com/watch?v=dQw4w9WgXcQ
$ INSERT www.en.wikipedia.org/wiki/Data_structure
$ INSERT www.wikihow.com/Deal-With-Tons-of-Homework
$ INSERT www.susi.uni-sofia.bg
$ SORT URL
$ PRINT
> www.en.wikipedia.org/wiki/Data_structure 150700011
www.susi.uni-sofia.bg 163750320
www.wikihow.com/Deal-With-Tons-of-Homework 150787241
www.youtube.com/watch?v=dQw4w9WgXcQ 150676023
$ SORT TIME
$ PRINT
>www.youtube.com/watch?v=dQw4w9WgXcQ 150676023
www.en.wikipedia.org/wiki/Data_structure 150700011
www.wikihow.com/Deal-With-Tons-of-Homework 150787241
www.susi.uni-sofia.bg 163750320

```