

Collections

Задача 1. Pager.

Да се направи клас, който получава списък от елементи и го разделя на страници. Дефиниран е размер на страницата (pageSize) – по колко елемента от списъка се показват наведнъж.

Класът има следните методи:

next() – връща следващите няколко елемента от списъка;

previous() – връща предишните няколко елемента на списъка;

hasNext() – връща дали има следващи елементи;

hasPrevious() – връща дали има предишни елементи;

firstPage() – връща първата страница и я прави текуща;

lastPage() – връща последната страница и я прави текуща;

getCurrentPageNumber() – връща номера на текущата страница.

Допълнителни условия:

При стартиране на програмата да се показва първата страница от списъка чрез извикване на next(). Методите next() и previous() да могат да се извикват от конзолата.

Когато е показана първата страница (първите няколко елемента от списъка) и се извика метода previous () – да се изписва съобщение за грешка.

Ако последната страница на списъка е с по-малко от pageSize елементи и се извика метода next () да **НЕ** се получава грешка.

При стартиране на програмата да може да се въведе от конзолата броя на елементите на страницата и самите елементи разделени с ,

Задача 2.

Да се направи програма, която брои по колко пъти се среща дадена дума в текст. Предминателни знаци не са думи.

Пр.

Hello, John. How are you John. Hello, Marry.

Hello – 2

John – 2

How – 1

are – 1

you – 1

Marry – 1

Задача 4. Да се направи Web Crawler.

Да се имплементира програма, която взема url, проверява го за валидност с регулярен израз. Ако е валиден то смъква html стараницата. Взема всички hyperlinks от страницата и ги проверява по същият начин. Действието се повтаря до дълбочина n (въвежда се). Програмата генерира файл в който са адресите и броя линкове, които са retrieve-нати от съответната страница. Ако един URL е вече парснат, то той не се парсва отново.

Hint – да се ползва Apache HttpClient

Пример:

dir.bg (23)

dnes.dir.bg (4)