**Lab03 - Kruhová fronta**

[Template domácího úkolu je ZDE](https://cw.fel.cvut.cz/wiki/_media/courses/b0b36pjv/hw/pjv-lab03.zip)

Napište program, který bude reprezentovat cyklickou frontu uchovávající hodnoty typu String. Kapacita fronty bude parametrem konstruktoru. Pokud bude použit bezparametrický konstruktor, vytvořte frontu o **konstantní velikosti 5**. Dále naimplementujte metody do připravené třídy CircularArrayQueue. Co mají jednotlivé metody dělat dozvíte v dokumentaci v interface Queue.java. Všimněte si, že dokumentace některých metod je velmi podobná (někdy i stejná) dokumentaci metod ve třídě [Queue](https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html) ve standardním java frameworku. Toto není náhoda, autoři úkolu se tímto javadocem inspirovali :-).

Odevzdávejte pouze soubor CircularArrayQueue.java.

Frontu implementujte **pomocí pole statické délky** se dvěma indexy ukazujícími na začátek a konec pole. Bližší informace naleznete na [wikipedii](https://en.wikipedia.org/wiki/Circular_buffer). **Jiná řešení nemusí být přijata**, v takových případech Vám cvičící po kontrole odevzdaného kódu sníží bodový zisk v BRUTE na 0.

Součástí hodnocení může být i namátkové manuální subjektivní hodnocení kvality kódu cvičícím. Dejte si tedy záležet.

Ve třídě Start je připraven kód, na kterém můžete funkčnost implementace kruhové fronty vyzkoušet

--- Příklad očekávaného výstupu programu

size: 6

value dequeued from CircularArrayQueue: Starkiller

printing all elements:

C-3PO

Jabba the Hutt

HK-47

Darth Nihilus

Count Dooku

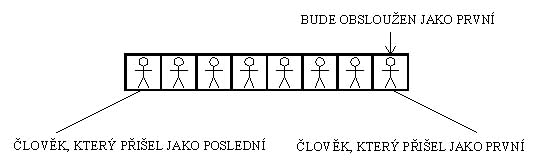
size: 6

**Fronta**

Fronta je dynamická množina (datová struktura), u které jsou specificky definovány operace výběru a vložení prvku. Operace výběr z fronty vybere prvek, který jsme vložili do fronty jako první. Při vkládání prvků do fronty se vkládaná položka vloží na jeho konec. (anglicky enqueue a dequeue)

Tato struktura se také někdy označuje termínem FIFO (first-in first-out).

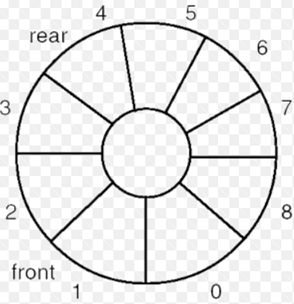
Fronta se dá implementovat polem a to buď polem statické délky s explicitním omezením na počet vložených prvků nebo polem dynamické délky. Alterantivně se dá také realizovat datovou strukturou nazývanou spojový seznam, se kterou se seznámíme na dalším cvičení.

[](https://cw.fel.cvut.cz/wiki/_detail/courses/b6b36pjv/hw/fronta.jpg?id=courses%3Ab0b36pjv%3Ahw%3A03)

Doplňující informace na [wiki](https://en.wikipedia.org/wiki/Queue_(abstract_data_type))

**Kruhová fronta**

V případě omezené kapacity fronty, například realizované polem statické délky, můžeme využít takzvanou kruhovou frontu. V té se začátek fronty pohybuje po jednotlivých prvcích pole tak jak jsou postupně prvky vkládány a odebírány. Praktické použití takové fronty si můžeme představit v případech, kdy do fronty jsou dávány jednotlivé požadavky na obsloužení, které v průměru nepřicházejí častěji než je rychlost obsluhy. V případě, že obsloužení konkrétního požadavku trvá déle, jsou další požadavky uloženy ve frontě a po vyřízení náročnějšího požadavku jsou pak ostatní, méně náročné, požadavky obslouženy rychleji a fronta je rychle vyprázdněna. Analogickou situaci můžeme začít například v obchodě, ve kterém je kapacita fronty omezena velikostí obchodu.

[](https://cw.fel.cvut.cz/wiki/_detail/courses/b6b36pjv/hw/cyklickafronta.png?id=courses%3Ab0b36pjv%3Ahw%3A03)

Kruhová fronta může explicitně hlídat, zda-li je možné nový požadavek do fronty vložit a v případě zaplnění fronty je možné požadavek na vložení zamítnout. Na druhé straně můžeme také najít připady, kdy do kruhové fronty dáváme požadavky a pokud je plná, jsou ty nejstarší požadavky přepisovány. To může například nastat v případě zpracování senzorických dat, ve kterém předpokládáme, že pokud se data nestihla zpracovat, jsou pravděpodobně již zastaralá a dáváme přednost zpracování aktuálnějších informací.