

SEMESTRÁLNÍ PRÁCE A

Maximální možný bodový zisk : **4 body**

A) V rámci modulu **ABSTRDOUBLELIST** implementujte abstraktní datovou strukturu (ADS): **obousměrně cyklicky zřetěžený lineární seznam** (bez hlavy) v dynamické paměti (stylizovaně znázorněný v rámci obr. 1). Tento modul implementuje následující rozhraní:

```
void Zrus() - zrušení celého seznamu,
boolean jePrazdy() - test naplněnosti seznamu,

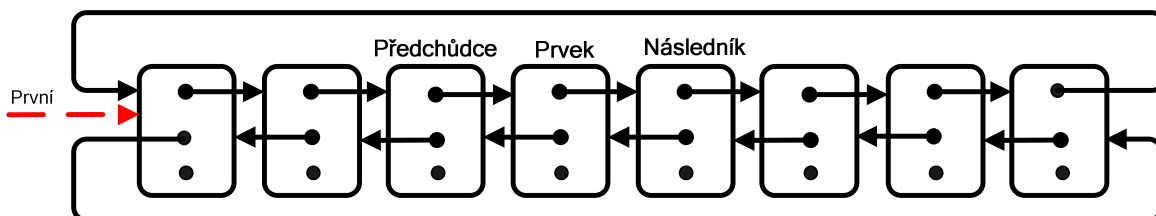
void vložPrvni(Object obj) - vložení prvku do seznamu na první místo
void vložPosledni(Object obj) - vložení prvku do seznamu na poslední místo,
void vložNaslednika(Object obj) - vložení prvku do seznamu jakožto následníka
aktuálního prvku,
void vložPredchudce(Object obj) - vložení prvku do seznamu jakožto předchůdce
aktuálního prvku,

Object zprístupniAktualni() - zpřístupnění aktuálního prvku seznamu,
Object zprístupniPrvni() - zpřístupnění prvního prvku seznamu,
Object zprístupniPosledni() - zpřístupnění posledního prvku seznamu,
Object zprístupniNaslednika() - zpřístupnění následníka aktuálního prvku,
Object pristupniPredchudce() - zpřístupnění předchůdce aktuálního prvku,

Object odeberAktualni() - odebrání (vyjmutí) aktuálního prvku ze seznamu,
aktuální prvek je nastaven na první prvek
Object odeberPrvni() - odebrání prvního prvku ze seznamu,
Object odeberPosledni() - odebrání posledního prvku ze seznamu,
Object odeberNaslednika() - odebrání následníka aktuálního prvku ze seznamu,
Object odeberPredchudce() - odebrání předchůdce aktuálního prvku ze seznamu,

Iterator vytvorIterator() - vytvoří iterátor
```

Pozn.: Operace typu **Vlož**, resp. **Odeber** necht' neprovádějí alokaci, resp. dealokaci prvku, na který se vztahují.



Obr.1 Obousměrně cyklicky zřetěžený lineární seznam (bez hlavy)

B) Pro ověření funkčnosti implementované ADS *obousměrně cyklicky zřetězený lineární seznam* (bez hlavy) v dynamické paměti vytvořte modul **SeznamOsob**, který implementuje následující rozhraní:

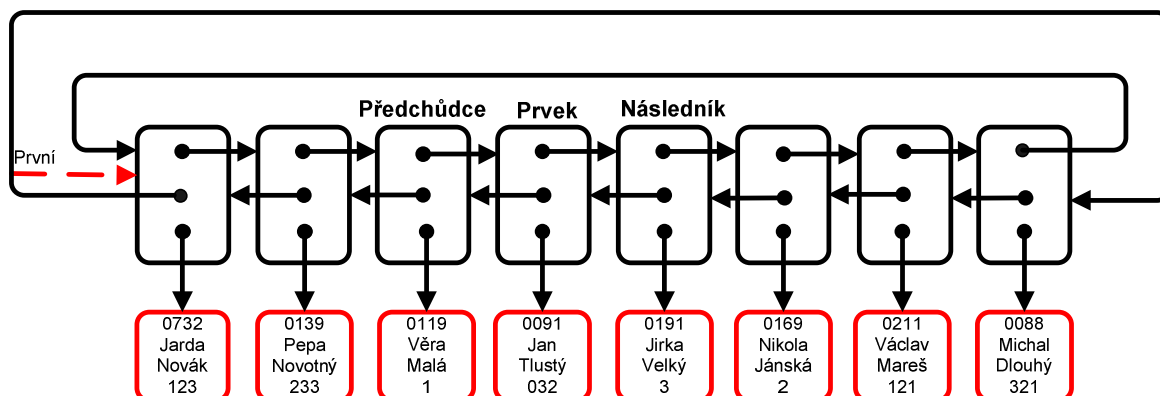
`void vloz(IOsoba osoba)` – vložení daného osoby do struktury (pokud ji tato již neobsahuje) na poslední místo seznamu,

`IOsoba najdi(IOsoba key)` – vyhledání osoby ve struktuře a uplatnění heuristiky *move-to-front*,

`IOsoba odeber(IOsoba osoba)` – odebrání definovaného osoby ze struktury.

Heuristika *moveToFront* je využívána k průběžnému uspořádávání struktury podle frekvence přístupu k jednotlivým prvkům. Podstatou zmíněné heuristiky je, že nalezený prvek (prostřednictvím operace *najdi*) je vždy přesunut na začátek příslušné struktury – důsledkem je tedy koncentrace často vyhledávaných prvků blízko začátku struktury.

Pozn.: Pro demonstraci funkčnosti *move-to-front* heuristiky je vhodné pro každý prvek struktury *SeznamOsob* zřídit čítač, jehož aktuální hodnota vyjádří počet dosud realizovaných přístupů k tomuto prvků.



Obr. 2 Stylizované znázornění datové struktury *SeznamOsob*

Modul **SeznamOsob** pracuje s typem osob:

- **vyučující** s informacemi *id, jméno, příjmení, číslo kanceláře*,
 - **student** s informacemi *id, jméno, příjmení, ročník*,
- přičemž jednotlivé osoby implementují rozhraní `IOsoba`.

C) Pro obsluhu aplikace vytvořte uživatelské rozhraní **ProgSeznamOsob**.

Zmíněný program necht' umožňuje zadávání vstupních dat z klávesnice, ze souboru a z generátoru, výstupy z programu necht' je možné zobrazit na obrazovce a uložit do souboru.