

## SEMESTRÁLNÍ PRÁCE A

Maximální možný bodový zisk: **4 body**

A) Motivační příklad:

V rámci personální agendy jsou uchovávány statistické informace o počtu obyvatel (muži/ženy) pro jednotlivé obce, přičemž jsou obce logicky organizovány do jednotlivých krajů. Finální implementace bude realizována jako pole abstraktních lineárních seznamů.

B) Použité datové struktury:

V rámci modulu **ABSTRDOUBLELIST** implementujte abstraktní datovou strukturu (ADS) **obousměrně cyklicky zřetěžený lineární seznam** v dynamické paměti (stylizovaně znázorněný v rámci obr. 1). Tato třída implementuje rozhraní `IAbstrDoubleList`, které implementuje implicitní rozhraní `Iterable`. Rozhraní `IAbstrDoubleList` je definováno následovně:

`void zrus()` –zrušení celého seznamu,

`boolean jePrazdny()` –test naplněnosti seznamu,

`void vložPrvni(T data)` –vložení prvku do seznamu na první místo

`void vložPosledni(T data)` –vložení prvku do seznamu na poslední místo,

`void vložNaslednika(T data)` –vložení prvku do seznamu jakožto následníka aktuálního prvku,

`void vložPredchudce(T data)` –vložení prvku do seznamu jakožto předchůdce aktuálního prvku,

`T zprístupniAktualni()` –zpřístupnění aktuálního prvku seznamu,

`T zprístupniPrvni()` –zpřístupnění prvního prvku seznamu,

`T zprístupniPosledni()` –zpřístupnění posledního prvku seznamu,

`T zprístupniNaslednika()` –zpřístupnění následníka aktuálního prvku,

`T zprístupniPredchudce()` –zpřístupnění předchůdce aktuálního prvku,

Pozn. Operace typu `zpřístupni`, přenastavují pozici aktuálního prvku

`T odeberAktualni()` –odebrání (vyjmutí) aktuálního prvku ze seznamu poté je aktuální prvek nastaven na první prvek

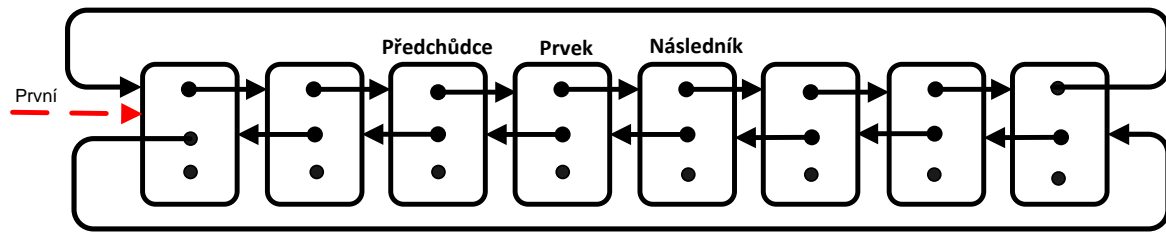
`T odeberPrvni()` –odebrání prvního prvku ze seznamu,

`T odeberPosledni()` –odebrání posledního prvku ze seznamu,

`T odeberNaslednika()` –odebrání následníka aktuálního prvku ze seznamu,

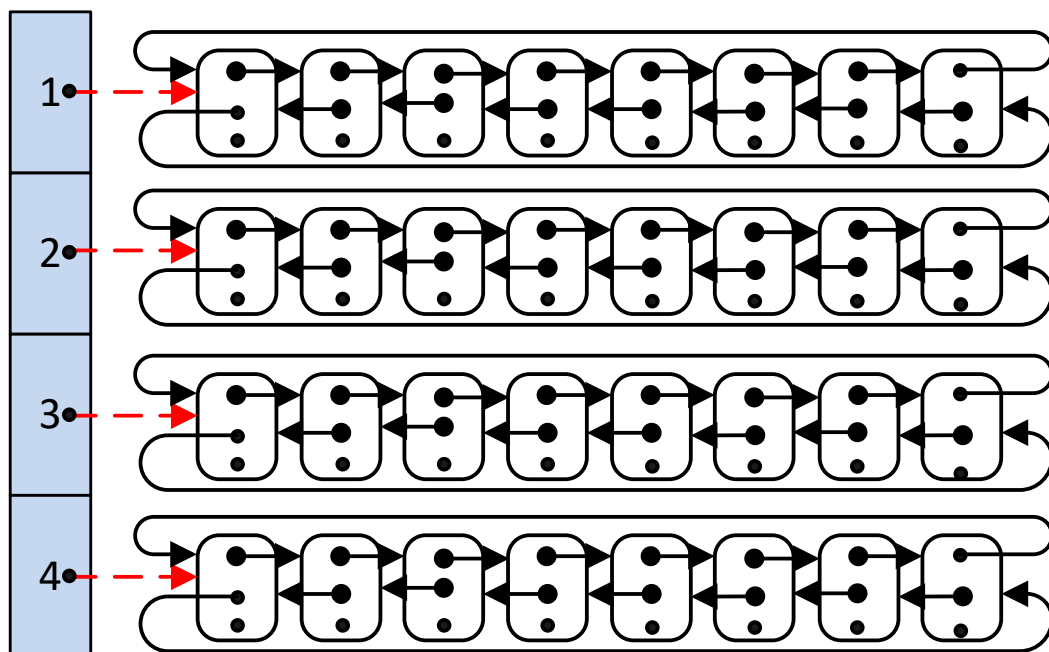
`T odeberPredchudce()` –odebrání předchůdce aktuálního prvku ze seznamu,

`Iterator<T> iterator()` –vytvoří iterátor (dle rozhraní `Iterable`)



Obrázek 1: Obousměrně cyklicky zřetěžený lineární seznam

Abstraktní lineární seznam slouží pro uchovávání jednotlivých obcí v daném kraji. Dále bude aplikace využívat datovou strukturu pole, které bude reprezentovat jednotlivé kraje a fixovat tak reference na jednotlivé obce v kraji, viz obr 2.



Obrázek 2: Pole seznamů

- C) Pro ověření funkčnosti implementovaných ADS vytvořte modul **Obyvatele**. Tento modul umožňuje správu seznamů obcí a implementuje následující rozhraní:

`int importData(String soubor)` - provede import dat z datového souboru kraje.csv, kde číslo kraje odpovídá indexu pole-1. Návrátová hodnota představuje počet úspěšně načtených záznamů.

`void vlozObec(Obec obec, enumPozice pozice, enumKraj kraj)` - vloží novou obec do seznamu obcí na příslušnou pozici (první, poslední, předchůdce, následník), v odpovídajícím kraji

`Obec zprístupniObec(enumPozice pozice, enumKraj kraj)` - zpřístupní obec z požadované pozice (první, poslední, předchůdce, následník, aktuální), v odpovídajícím kraji

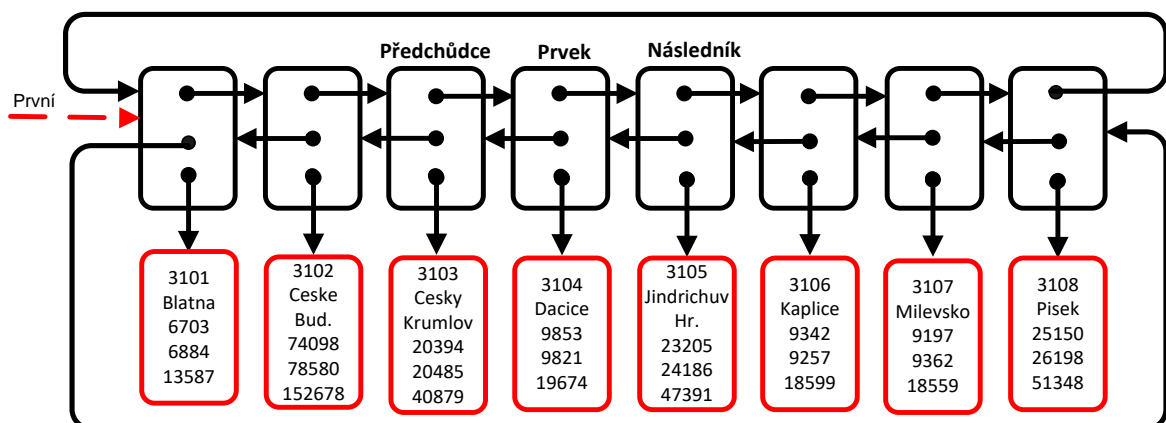
`Obec odeberObec(enumPozice pozice, enumKraj Kraj)` – odebere obec z požadované pozice (první, poslední, předchůdce, následník, aktuální), v odpovídajícím kraji

`float zjistiPrumer(enumKraj Kraj)` – zjistí průměrný počet obyvatel v kraji, pokud je hodnota kraje rovna null, pak je průměr spočítán pro všechny kraje.

`Obec[] zobrazObce(enumKraj Kraj)` – pomocí iterátoru provede výpis obcí v daném kraji, pokud je hodnota kraje rovna null, pak jsou vypsány všechny kraje. Alternativně může metoda vrátit `IAbstrDoubleList`

`Obec[] zobrazObceNadPrumer(enumKraj Kraj)` – pomocí iterátoru provede výpis obcí, které mají v daném kraji nadprůměrný počet obyvatel. Pokud je hodnota kraje rovna null, pak je průměr spočítán pro všechny kraje. Alternativně může metoda vrátit `IAbstrDoubleList`

`void zrus(enumKraj Kraj)` – zruší všechny obce v kraji. Pokud je hodnota kraje rovna null, pak zruší všechny obce.



Obrázek 3: Sylyzované znázornění seznamu obcí ve vybraném kraji

Modul **Obec** pracuje s typem

- PSC
- Obec
- Počet mužů
- Počet žen
- celkem

D) Pro obsluhu aplikace vytvořte uživatelské **formulářové** rozhraní **ProgObyvatele**, které umožňuje obsluhu programu a volat požadované operace.

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.