UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

INFORMAČNÍ TECHNOLOGIE

S E M E S T R Á L N Í P R Á C E

z předmětu

Datové Struktury

Semestrální práce A

Autor: Jiří Hermann

Obsah

[1. Úvod 3](#_Toc351496059)

[Téma : Obousměrně cyklicky zřetězený lineární seznam 3](#_Toc351496060)

[2. Výhody a nevýhody 4](#_Toc351496061)

[Výhody 4](#_Toc351496062)

[Nevýhody 4](#_Toc351496063)

[3. Problémy s implementací 5](#_Toc351496064)

[4. Agregace a Dekompozice 6](#_Toc351496065)

[Agregace : 6](#_Toc351496066)

[spojování, seskupování, shlukování 6](#_Toc351496067)

[Dekompozice: 6](#_Toc351496068)

[rozložení, rozklad 6](#_Toc351496069)

[5. Seznam Obrázků 7](#_Toc351496070)

[6. Použité odkazy 8](#_Toc351496071)

[7. Závěr 9](#_Toc351496072)

# Úvod

## Téma : Obousměrně cyklicky zřetězený lineární seznam

Každý prvek obsahuje odkaz na následující i předchozí prvek. Seznam je tedy možné seznam procházet dvěma směry. Při vložení prvního prvku do datové struktury kdy je datová struktura prázdná nastavíme reference předchozího a následníka na vkládaný prvek. Náš lineární seznam funguje cyklicky kdy můžeme projít celý seznam jak pomocí reference na následující prvek tak i pomocí reference na předchozí prvek.

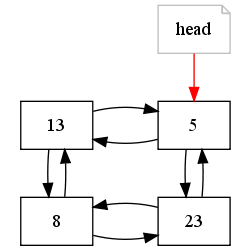
# Výhody a nevýhody

## Výhody

Na rozdíl od pole nemusí prvky následovat v paměti lineárně za sebou a tak se může počet prvků dynamicky rozrůstat bez potřeby realokovat paměťový prostor. Při zvětšování seznamu stačí vytvořit nový prvek a spojit jej pomocí ukazatelů s ostatními.

## Nevýhody

Právě ona „roztříštěnost“ seznamu, kdy je každý prvek na libovolném místě v paměti, zvyšuje časovou náročnost některých operací (např. vyhledávání, vkládání, mazání). Seznam je totiž nutné často procházet celý, protože neznáme přesná umístění všech prvků. Také implementace seznamu je složitější, protože je nutné pracovat korektně s ukazateli.



Obrázek 1-Obousměrný cyklicky zřetězený lineární seznam

# Problémy s implementací

Problémy s implementací nastali při vytvoření fronty kam jsme ukládali prvky bud pro agregaci nebo pro dekompozici. Hlavní jádrem problému bylo si uvědomit jaká data budeme do fronty vkládat a jak je budeme odebírat a dále vkládat po reorganizaci do ADL.

**Metoda Vytipuj :**

metodu vytipuj kandidáty kde u možnosti Agregace musí být splněna podmínka že prvek který splňuje časové kritérium musí mít i následující prvek který toto kritérium splňuje. Z fronty z ADL vybírám 2 prvky a ty také vkládám do fronty. Prvek si drží své reference následující a předchozí. U dekompozice vkládám jenom jeden prvek.

**Metoda reorganizuj :**

V metodě reorganizuj bylo nejtěžší porovnat prvky a poté je vložit na místo v ADL tam kde opravdu měli být. Zajistil jsem si to pomocí zjištění prvku iterátoru který se musí rovnat prvku ve frontě který odebírám a když projde porovnáním tak bud rozděluji nebo spojuji procesy a následně vkládám do ADL.

# Agregace a Dekompozice

## Agregace :

Význam:

## spojování, seskupování, shlukování

V naše případě spojujeme dva vytipované prvky do jednoho.

## Dekompozice:

Význam:

## rozložení, rozklad

V našem případe rozkládáme jeden vytipovaný prvek na dva prvky.

# Seznam Obrázků

[Obrázek 1-Obousměrný cyklicky zřetězený lineární seznam 4](#_Toc351368877)

# Použité odkazy

1. <http://slovnik-cizich-slov.abz.cz/web.php/slovo/dekompozice-dekomposice>
2. <http://voho.cz/wiki/informatika/datova-struktura/seznam/>
3. <http://slovnik-cizich-slov.abz.cz/web.php/slovo/agregace>

# Závěr

V semestrální práci jsme měli za úkol implementovat ADL. V semestrální práci jsem se naučil pracovat s genericitou a iterátorem, který byl stěžejní věc pro naší práci. Slouží nám jako průvodce naší datovou strukturou a vrací nám dané prvky. V příloze také najdete PDF dokument kde naleznete UML diagram Semestrální práce A.