

ADSO

5. Vyhledávací posloupnost

(prezentace k učebnici)

Ivan Ryant

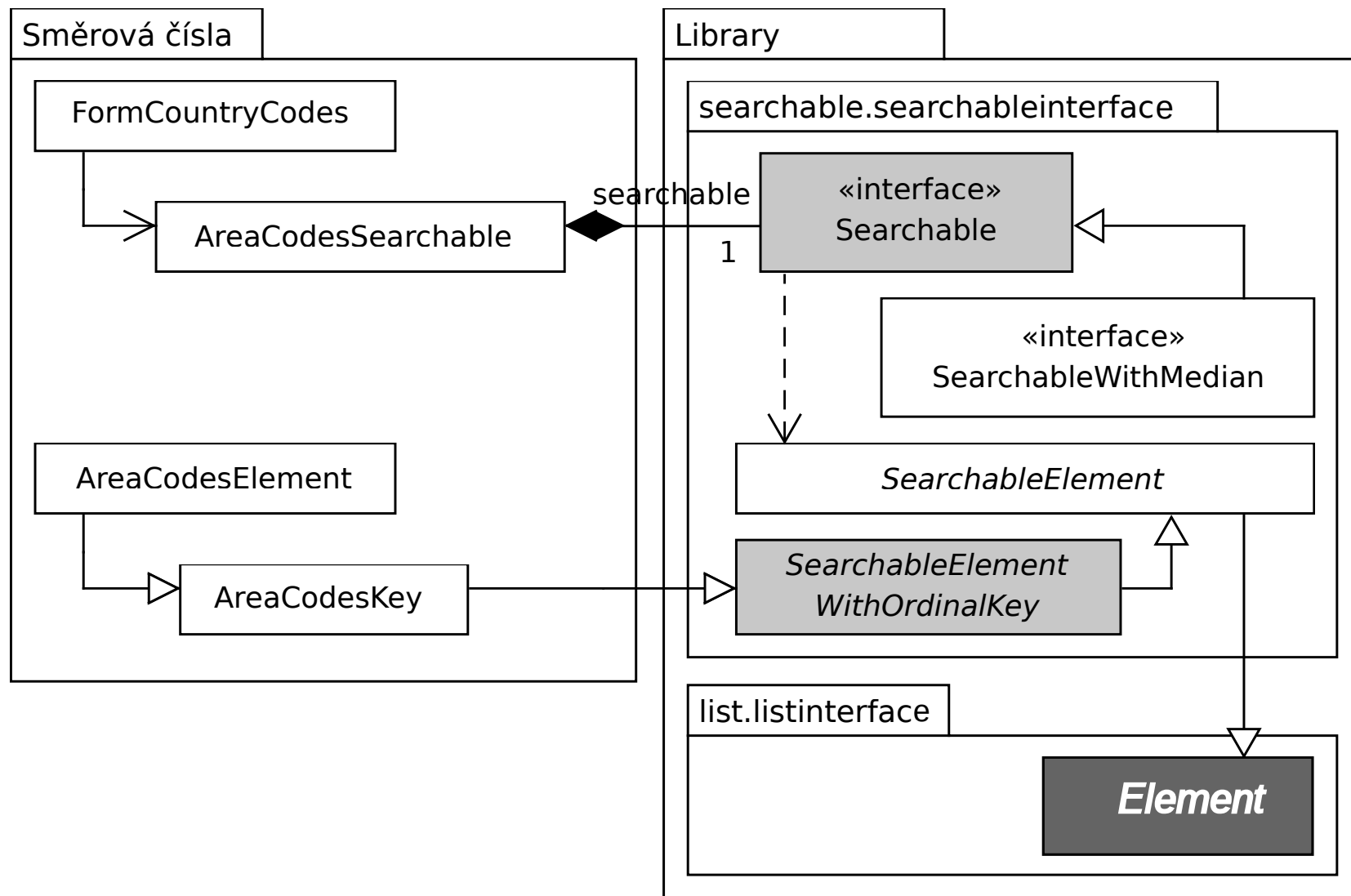
Agenda

- Základní pojmy
- Úloha abstraktních datových typů
- Posloupnosti a operace s nimi
- Vyhledávací datové struktury
- **Vyhledávací posloupnost**
- Algoritmy řazení
- Stromy
- Rozptýlené tabulky
- Prohledávání do hloubky a do šířky
- Práce s grafy
- Techniky návrhu efektivních algoritmů

Agenda

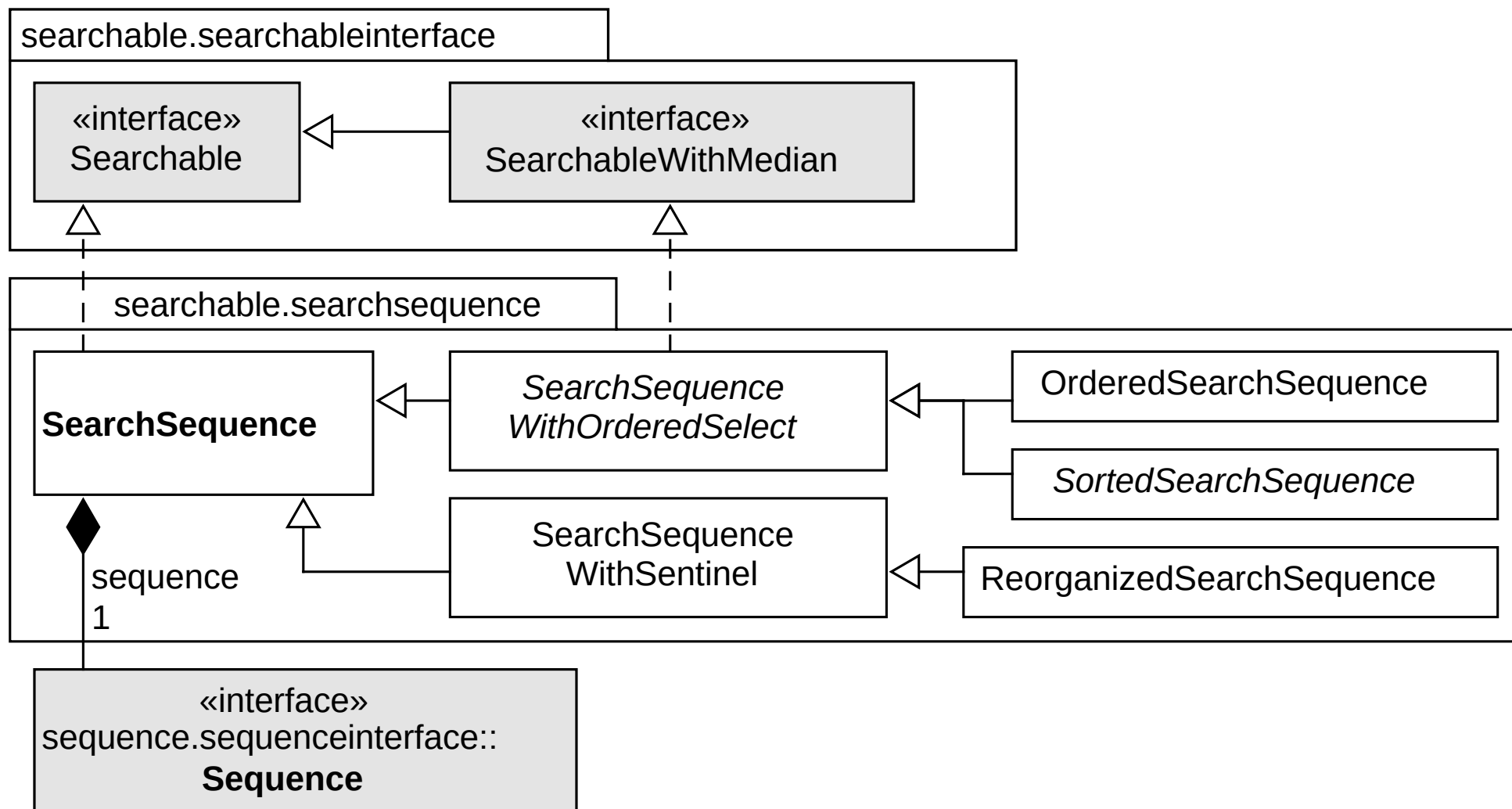
- Základní pojmy
- Úloha abstraktních datových typů
- Posloupnosti a operace s nimi
- Vyhledávací datové struktury
- **Vyhledávací posloupnost**
 - Obyčejná vyhledávací posloupnost
 - Vyhledávání s pomocí zarážky
 - Reorganizovaná posloupnost
 - Uspořádaná posloupnost, medián
 - Seřazená posloupnost
 - Shrnutí
- Algoritmy řazení
- Stromy
- Rozptýlené tabulky
- Prohledávání do hloubky a do šířky
- Práce s grafy
- Techniky návrhu efektivních algoritmů

Vyhledávací posloupnost



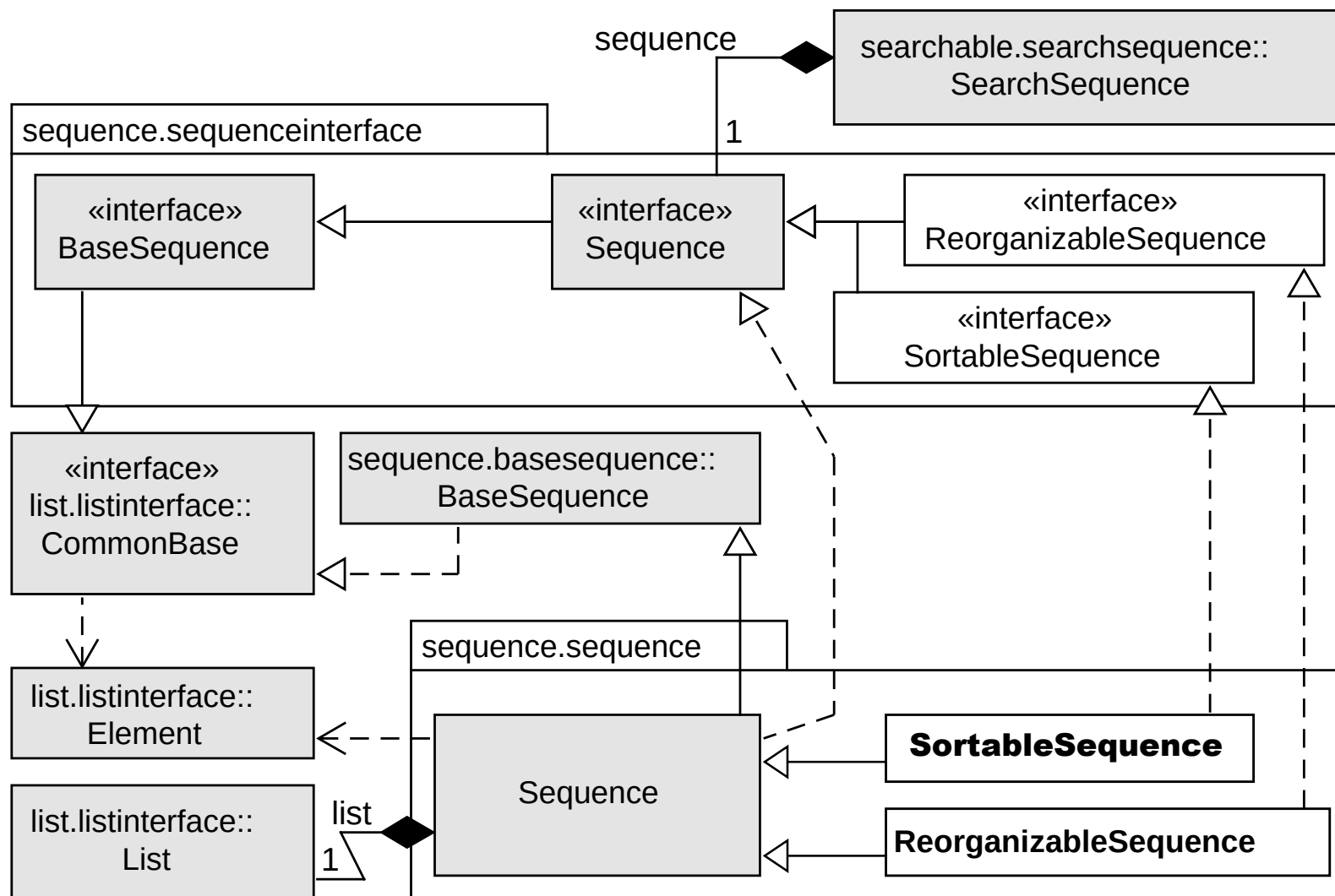
Design aplikace „Směrová čísla“

Vyhledávací posloupnost



Design různých druhů vyhledávacích posloupností

Vyhledávací posloupnost



Design různých druhů posloupností vhodných k vyhledávání

Vyhledávací posloupnost

- **Obyčejná vyhledávací posloupnost**
- Vyhledávání s pomocí zářky
- Reorganizovaná posloupnost
- Uspořádaná posloupnost, medián
- Seřazená posloupnost
- Shrnutí

Obyčejná vyhledávací posloupnost

- Metody vyhledávací posloupnosti

- Insert
- Append
- Remove

se velmi jednoduše mapují na metody komponenty `sequence`.

- Metoda `SearchSequence.Select`

- Prochází posloupnost prvek po prvku, dokud nenarazí na hledaný prvek (testuje se shoda klíčů) nebo na konec posloupnosti.
- Na konci metody se pak musí odlišit, jestli prvek byl nalezen, nebo ne.

Vyhledávací posloupnost

- Obyčejná vyhledávací posloupnost
- **Vyhledávání s pomocí zářky**
- Reorganizovaná posloupnost
- Uspořádaná posloupnost, medián
- Seřazená posloupnost
- Shrnutí

Vyhledávání s pomocí zářáčky

- Metoda `SearchSequence.Select`
 - opakovaně testuje
 - shodu klíčů
 - konec posloupnosti
 - Při odstraňování posledního prvku z jednosměrně zřetěženého seznamu se musí projít celý seznam od začátku do konce.
- Metoda `SearchSequenceWithSentinel.Select`:
 - Za poslední prvek připojíme zářáčku vybavenou hledaným klíčem – pak stačí testovat jen shodu klíčů.
 - Zjednodušení testu zrychlí výpočet.
 - Poslední prvek z jednosměrně zřetěženého seznamu se odstraňuje stejně efektivně jako každý jiný prvek.

Vyhledávací posloupnost

- Obyčejná vyhledávací posloupnost
- Vyhledávání s pomocí zarážky
- **Reorganizovaná posloupnost**
- Uspořádaná posloupnost, medián
- Seřazená posloupnost
- Shrnutí

Reorganizovaná posloupnost

- Účel
 - Zrychlit prohledávání
- Princip řešení
 - Prvek, který se chystáme hledat, umístit dopředu
- Jak to udělat?
 - Heuristicky (zrychlení není zaručeno), např.
 - Nejčastěji hledané prvky posouvat k začátku
 - Pracnější, ale funguje to v mnoha různých úlohách
 - Naposledy nalezený prvek umístit do čela
 - Snazší, ale funguje jen v uváženě vybraných úlohách
 - Tato strategie se implementuje v příkladu

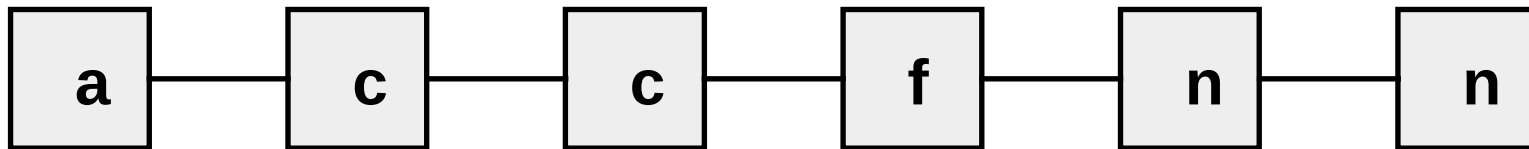
Reorganizovaná posloupnost

- Implementace:
 - Ve vyhledávací struktuře `SearchSequence`
 - Redefinujeme metody
 - `Select` (použije `sequence.MakeFirst`)
 - `Update`
 - Ve vložené posloupnosti
`sequence: ReorganizableSequence`
 - Přidáme metodu
 - `MakeFirst`

Vyhledávací posloupnost

- Obyčejná vyhledávací posloupnost
- Vyhledávání s pomocí zarážky
- Reorganizovaná posloupnost
- **Uspořádaná posloupnost, medián**
- Seřazená posloupnost
- Shrnutí

Uspořádaná a seřazená posloupnost



- *SearchSequenceWithOrderedSelect*

Je to abstraktní třída s abstraktní metodou `SelectMedian`

Třídy odvozené děděním:

- `OrderedSearchSequence`

- Uspořádaná posloupnost
- Je vždy uspořádaná, prvky se vkládají na správné místo metodou `Insert`
- Je efektivní při častém hledání a řídkém vkládání

- `SortedSearchSequence`

- Seřazená posloupnost
- Seřadí se (`Sort`) vždy až před hledáním (`Select`)
- Je efektivní při častém vkládání ve shlucích

Medián

- Medián je přibližně prostřední prvek v uspořádané posloupnosti
 - Je-li počet prvků lichý, je medián přesně uprostřed
 - Je-li počet prvků sudý, není medián určen jednoznačně
 - V metodě `OrderedSearchSequence.SelectMedian` počítám umístění mediánu takto:

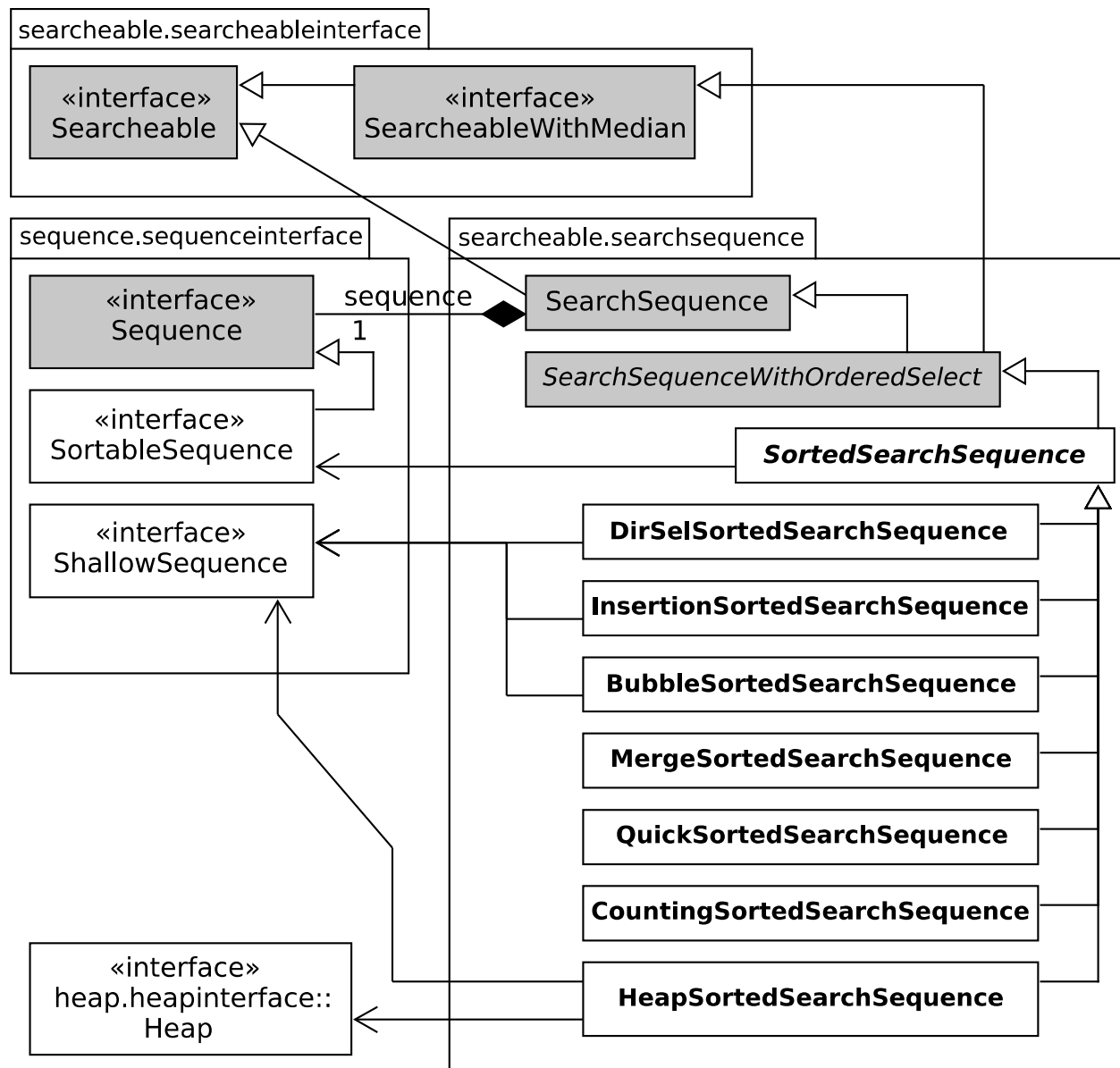
`sequence.Count div 2`

- Kde `Count` zjistí počet prvků v posloupnosti
- První prvek posloupnosti je umístěn na indexu 0

Vyhledávací posloupnost

- Obyčejná vyhledávací posloupnost
- Vyhledávání s pomocí zarážky
- Reorganizovaná posloupnost
- Uspořádaná posloupnost, medián
- **Seřazená posloupnost**
- Shrnutí

Seřazená posloupnost



Předkové a potomci třídy `SortedSearchSequence`

Seřazená posloupnost

- abstraktní třída `SortedSearchSequence`
 - je společným předkem několika odvozených tříd
 - abstraktní metoda `Sort` je v odvozených třídách implementována různě
 - různé algoritmy řazení
 - před vykonáním zděděných metod `Select` a `Remove` musí být posloupnost seřazena
 - datová položka `sequence` je typu `SortableSequence`
 - dá se mělce klonovat (tím se dá např. rozdělit na seřazenou a neseřazenou část)

Mělká posloupnost

- Slouží ke klonování posloupnosti
 - Prvky klonů se sdílejí, nekopírují se
 - Viz rozhraní `ShallowSequence`
- POZOR: Moje implementace není vzorová, univerzální, opakovaně použitelná
 - Je to účelová pomůcka pro omezené užití
- Alternativa:
 - Návrhový vzor iterátor
 - Je komplikovanější
 - Viz kurs objektového designu

Vyhledávací posloupnost

- Obyčejná vyhledávací posloupnost
- Vyhledávání s pomocí zarážky
- Reorganizovaná posloupnost
- Uspořádaná posloupnost, medián
- Seřazená posloupnost
- **Shrnutí**

Shrnutí

- **Obyčejná vyhledávací posloupnost** je jednoduchá a snadno se programuje.
- **Zarážka** nám nejen pomůže trochu zjednodušit a zrychlit vyhledávání, ale především nám umožní zjednodušit a zrychlit odstraňování posledního prvku z jednosměrně zřetězeného spojového seznamu.
- **Reorganizovaná posloupnost** může zrychlit vyhledávání přímo zázračně. Zrychlení však bohužel není zaručeno, protože je založeno na **heuristice**.
- Zaručeného a významného zrychlení lze dosáhnout mechanismem *bisekce* na **uspořádané** nebo **seřazené** posloupnosti. Uspořádání nebo seřazení posloupnosti se hodí také k rychlému slévání posloupností např. při implementaci množinových sjednocení, průniků apod. operací.

Konec

Tato prezentace patří k učebnici *Algoritmy a datové struktury objektově* od Ivana Ryanta. Obsahuje texty a obrázky z této učebnice.

Tato prezentace smí být volně šířena, ale vždy i s tímto snímkem. Citujete-li, vyznačte zřetelně citát v plném rozsahu a uveďte zdroj:

RYANT, Ivan. *Algoritmy a datové struktury objektově*. Praha: Ivan Ryant, 2017. ISBN 978-80-270-1660-0