

# Vektory, Seznamy a Sekvence

Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole

**Interreg**   
EVROPSKÁ UNIE  
**Rakousko-Česká republika**  
Evropský fond pro regionální rozvoj



**Europäische Union**  
**Evropská unie**  
Europäischer Fonds für  
regionale Entwicklung  
Evropský fond pro  
regionální rozvoj



**UNIVERSITY  
OF APPLIED SCIENCES  
UPPER AUSTRIA**

# Vektory - ADT

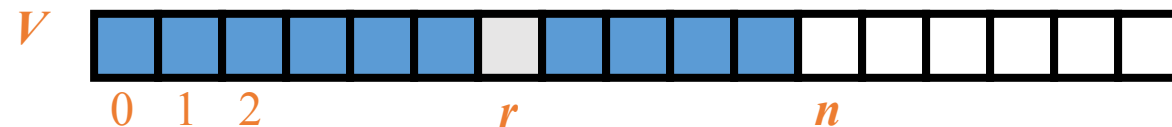
- Rozšiřují pojem pole ukládáním sekvence libovolných objektů
- Prvek může být čten, vkládán a odebírán pomocí určení jeho pořadí
- Operace s vektorem:
  - object elemAtRank (integer r)**
  - object replaceAtRank (integer r, object o)**
  - insertAtRank(integer r, object o)**
  - object removeAtRank(integer r)**
- Pomocné operace
  - integer size()**      **boolean isEmpty()**

# Vektory - ADT

- Výjimky
  - Nesprávný index prvku (záporný index...)
- Aplikace
  - Přímé
    - Tříděná kolekce objektů (základní databáze)
  - Nepřímé
    - Pomocná datová struktura pro algoritmy
    - Části jiných datových struktur

# Vektory – implementace pomocí pole

- Použití pole  $V$  o délce  $N$
- Proměnná  $n$  určuje délku vektoru
- Operace *isEmpty()*, *elemAtRank(r)*, *replaceAtRank(r, O)* - časová náročnost  $O(1)$
- Operace *insertAtRank(r, O)* - časová náročností  $O(n)$
- Operace *removeAtRank(r)* - časová náročností  $O(n)$



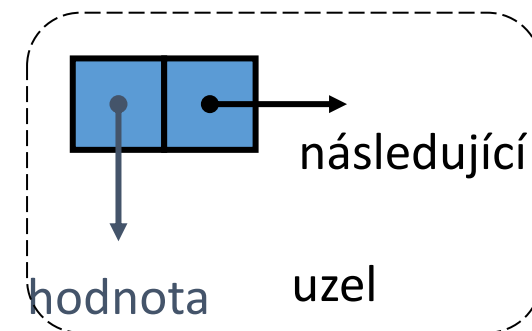
# Seznam – ADT

- Posloupnost pozic ukládajících libovolná data
- Zavádí vztahy před/po mezi pozicemi
- Obecné operace
  - `size()`, `isEmpty()`
- Dotazovací operace
  - `isFirst()`, `isLast()`
- Aktualizační operace
  - `replaceElement(p, o)`, `swapElements(p, q)`, `insertBefore(p, o)`, `insertAfter(p, o)`, `insertFirst(o)`, `insertLast(o)`, `remove(p)`
- Přístupové operace
  - `first()`, `last()`, `before()`, `after()`

# Seznam

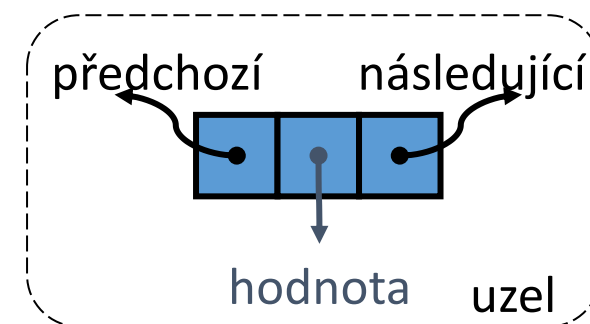
- Single linked list – jednosměrný spojový seznam

- Prvek obsahuje odkaz pouze na následující uzel
- Lze použít pro implementaci zásobníku ( $O(n)$  pro paměť,  $O(1)$  pro každou operaci) i fronty ( $O(n)$  pro paměť,  $O(1)$  pro každou operaci)



- Double linked list – obousměrný spojový seznam

- Prvek obsahuje odkaz na předcházející i následující uzel



# Sekvence – ADT

- Spojení Vektoru a Seznamu
- Přístup k prvkům pomocí pořadí nebo pozice
- Obecné operace
  - `size()`, `isEmpty()`
  - Propojující operace
    - `atRank(r)`, `rankOf(p)`
- Vektorové operace
  - `elemAtRank (r)`, `replaceAtRank (r, o)`, `insertAtRank(r, o)`, `removeAtRank(r)`
- Seznamové operace
  - `first()`, `last()`, `before()`, `after()`, `replaceElement(p, o)`, `swapElements(p, q)`, `insertBefore(p, o)`, `insertAfter(p, o)`, `insertFirst(o)`, `insertLast(o)`, `remove(p)`

# Sekvence

- Základní, obecně použitelný datový typ pro ukládání uspořádaného souboru prvků
- Aplikace
  - Přímé
    - Obecná náhrada za zásobník, frontu, vektor, nebo seznam
    - Malá databáze
  - Nepřímé
    - Stavební blok pro komplexnější datové struktury